



Java

A Complete Practical Solution

01 0 1

00 011

0101

1 1

01 0 1 00 011

1 00 011

0101

SWATI SAXENA



JAVA

A complete Practical Solution

Swati Saxena



www.bpbonline.com

FIRST EDITION 2018

SECOND REVISED & UPDATED EDITION 2021

Copyright © BPB Publications, India

ISBN: 978-93-88176-50-7

All Rights Reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by any means or stored in a database or retrieval system, without the prior written permission of the publisher with the exception to the program listings which may be entered, stored and executed in a computer system, but they can not be reproduced by the means of publication, photocopy, recording, or by any electronic and mechanical means.

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Distributors:

BPB PUBLICATIONS

20, Ansari Road, Darya Ganj
New Delhi-110002
Ph: 23254990 / 23254991

DECCAN AGENCIES

4-3-329, Bank Street,
Hyderabad-500195
Ph: 24756967 / 24756400

MICRO MEDIA

Shop No. 5, Mahendra Chambers,
150 DN Rd. Next to Capital Cinema,
V.T. (C.S.T.) Station, MUMBAI-400 001
Ph: 22078296 / 22078297

BPB BOOK CENTRE

376 Old Lajpat Rai Market,
Delhi-110006
Ph: 23861747

To View Complete
BPB Publications Catalogue
Scan the QR Code:



Published by Manish Jain for BPB Publications, 20 Ansari Road, Darya Ganj, New Delhi-110002 and Printed by him at Repro India Ltd, Mumbai
www.bpbonline.com

About the Author

Swati Saxena, Assistant professor in Computer Science is MCA, OCJP (Oracle Certified Java Professional), having in-depth knowledge of subject and very vast experience in developing and training.

She is well known Author of computer science books and has written books for BCA, MCA, BTech students and as per NCERT syllabus for XI-XII standard.

She has been awarded with following prestigious awards for her writing work:

1. Jaipur Ratan Award
2. Pratibha Samman
3. Rajasthan Women Achievement Award
4. Swami Vivekanand National Award

Her knowledge and teaching is always praised by her mentor as well as students.

Swati is author of “C programming and coding Question Bank with solution, Cracking Kotlin Interview, Kotlin At a Glance, Python Interview Questions”.

The alumnus of her, are well placed in many reputed organizations all over India.

Preface

This book helps you to understand each and every topic of java practically.

It will help you in developing software and websites because one should have sound practical knowledge.

It covers all the topics which are important from the point of view of the interview, certification and examinations and no topic is left untouched.

This is a complete practical book.

This book has been written after a long experience .

Despite the fact that ample care has been taken, the possibility of minor inaccuracies cannot be ruled out.

So, if any, your suggestions are highly solicited.

Lastly, Thanks to all the students who believe in me.

Swati Saxena

Acknowledgement

I am thankful of the BPB publication who inspire me and asked me to work on this book “Java-A complete Practical Solution”.

I thank Prof. S.C. Saxena, Ex-Principal, Maharishi Arvind Institute of Technology, Jaipur, who inspire me time to time. I thank Mr. Naveen Saxena for his constant support during the preparation of this book. I am thankful of Prof. Nidhi Saxena, Ex-Assistant professor PESIT University, Bangalore for her guidance and support.

Lastly, I thank the management, editorial and production staff of BPB publication, New Delhi for bringing out this book in record time.

Table of Contents

History in Brief.....	VIII
Magic Code : Bytecode	1
Operators in java	11
Java Comment.....	13
Java Control Statement	13
Iteration / Looping	19
Array	27
Object and classes.....	32
Constructor	40
Static.....	45
“This” Keyword.....	49
Final Keyword.....	52
Java Regular Expressions (RegeX).....	55
String.....	57
Instanceof	60
Inner Class.....	62
Inheritance	65
<i>Super</i>	65
<i>Overriding</i>	65
Abstraction.....	70
<i>Abstract Class</i>	70

<i>Interface</i>	72
<i>Exception</i>	75
<i>The try-with-resource statement</i>	79
Package	85
Collection and Generics	91
Applet.....	102
<i>Life Cycle of Java Applet</i>	102
<i>AWT package</i>	102
Adapter Class	130
Multithreading	148
Networking.....	162
File Handling (IO package).....	174
Serialization.....	178
Java Advance	182
<i>Swing</i>	182
<i>JApplet</i>	213
<i>JDBC</i>	222
<i>Servlet</i>	268
<i>JSP</i>	293
Extra Efforts	309
<i>RMI, Bor der in Swing Component , LookAndFeel etc</i>	309
Some Interview Question based on practical and Theory	339
Advance Java New Updates.....	374

HISTORY IN BRIEF:

Java is a general purpose, class based, object oriented, platform independent, portable, architecturally neutral, multithreaded, dynamic, distributed, portable, and robust interpreted programming language.

Java was originally designed for interactive television, but it was too advanced technology for the digital cable television industry at the time.

Java was started as a green project by James Gosling, Patrick Naughton, Chris Warth, Mike Sheridan, and Ed Frank at Sun Microsystems in June 1991.

Team members of green project starts this project to develop a language for digital devices such as set-top boxes, televisions, and so on. But, it was suited for internet programming.

Firstly it was called as *GreenTalk*. This language was initially called *oak* but was renamed as *Java* in 1995. It made the promise of *Write Once, Run Anywhere*, with free runtimes on popular platforms.

It was fairly secure and its security was configurable, allowing for network and file access to be limited.

KeyPoints:

Java is an island of Indonesia where the first coffee was produced (called java coffee).

- Java is just a name not an acronym.
- It was developed by James Gosling and team at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995.
- 1st version was JDK 1.0.

Versions of Java:

- JDK 1.0 (Januar 1996)
- J2SE 1.2 (December 8, 1998)
- J2SE 1.4 (February 6, 2002)
- Java SE 6 (December 11, 2006)
- Java SE 8 (March 18, 2014)
- Java SE 10 (March 20, 2018)
- JDK 1.1 (February 19, 1997)
- J2SE 1.3 (May 8, 2000)
- J2SE 5.0 (September 30, 2004)
- Java SE 7 (July 28, 2011)
- Java SE 9 (September 21, 2017)

MAGIC CODE : BYTECODE:

Java Byte Code is the language to which Java source is compiled and the Java Virtual Machine understands. Unlike compiled languages that have to be specifically compiled for each different type of computers, a Java program only needs to be converted to byte code once, after which it can run on any platform for which a Java Virtual Machine exists.

Bytecode is a set of instruction designed to be executed by the java run time system, which is called JVM.

JVM is an interpreter of bytecode.

Features of Java:

- ✓ Simple
- ✓ Secure
- ✓ Platform Independent
- ✓ Object Oriented
- ✓ Robust
- ✓ Portable
- ✓ Dynamic
- ✓ Multithreaded
- ✓ Compiled and interpreted
- both ✓ Supports networking
- ✓ High Performance
- ✓ Distributed
- ✓ Architecture Neutral

Requirement for Java Example

For executing any java program, you need to,

- install the JDK if you don't have installed it, download the JDK and install it.
- set path of the jdk/bin directory.
- create the Java program.
- compile and run the Java program

For basic, looping, conditional question, and example please refer : “C Programming and Coding, Question Bank with Solution by swati saxena (BpB)”

1. Write a simple java program to print a message?

```
class First
{
    public static void main(String [] arg)
    {
        System.out.println("Welcome to Swati Computers");
    }
}
```

Save it as : First.java

Compile: C:\>javac First.java (it will create a First.class file that contains bytecode)

Run: c:\> java First

Keywords in java:

Abstract	assert	boolean	break
byte	case	catch	char
class	const	continue	default
do	double	else	enum
extends	final	finally	float
for	goto	if	implements
import	instanceof	int	interface
long	native	new	package
private	protected	public	return

short	static	strictfp	super
switch	synchronized	this	throw
throws	transient	try	void
volatile	while	true	false
null			

Data Types in Java:

Data types represent the different values to be stored in the variable. In java, there are two types of data types:

- Primitive data types
- Non-primitive data types

Datatype

1. Primitive

A) Boolean

boolean

B) Numeric

Character

char

Integral

Integer

byte

short

int

long

Floating point

float

double

2. Non primitive

A) String

B) Array

C) etc.

Types of Variable:

There are three types of variables in Java:

- local variable
- instance variable
- static variable

2. Write a Java program to interchange values of two variable without using any third variable?

```
class Swap
{
    public static void main(String [] arg)
    {
        int a=2,b=3;
        a=a+b;
        b=a-b;
        a=a-b;
        System.out.println("A="+a+"\nb="+b);
    }
}
```

3. Write a Java program to check whether main() function can overload or not?

```
class Overload
{
    public static void main(Integer [] aq)
    {
        System.out.println("hi");
    }
    public static void main(String [] ar)
    {
        System.out.println("Hello");
    }
}
```

4. Write a program to input length and breadth of rectangle and calculate the area (Input through command line argument)?

```
class Rect_area
{
public static void main(String [] aa)
{
    int length= Integer.parseInt(aa[0]);
    int breath= Integer.parseInt(aa[1]);
    int area=length*breath;
    System.out.println("the output is "+area);
}
}
```

5. Write a program to input total money and check how many of two thousand, five hundred, hundred, fifty, twenty, ten, five rupee note we need?

```
import java.util.*;
class notes
{
    public static void main(String [] ar)
    {
        int thousand_2,hundred_5,hundred_1,fifty,ten,five,note;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter Rupees");
        note=sc.nextInt();

        thousand_2=note/2000;
        note=note%2000;

        hundred_5=note/500;
        note=note%500;

        hundred_1=note/100;
        note=note%100;
```

```
fifty=note/50;
note=note%50;
```

```
ten=note/10;
note%=10;
```

```
five=note/5;
note=note%5;
```

```
System.out.println("We need \n2000rs note:"+thousand_2+"\n500rs note:"+hundred_5+"\n100rs note:"+hundred_1+"\n50rs note:"+fifty+"\n10rs note:"+ten+"\n5 rs note:"+five+"\nexttra:"+note); }
```

```
}
```

6. Write a program to calculate the simple interest (input through command-line)?

```
class SI
{
    public static void main(String [] aa)
    {

        int principal= Integer.parseInt(aa[0]);
        int time= Integer.parseInt(aa[1]);
        double rate=Double.parseDouble(aa[2]);
        double si=(principal*time*rate)/100;
        System.out.println("the SI is "+si);
    }
}
```

7. Write a program to input the name and age of a person and print (input through Scanner class)?

```
import java.util.*;
class inp
```



```

{
public static void main(String [] ar)
{
    Scanner sc; //reference variable
    sc=new Scanner(System.in); //object
    String name;
    String lname;
    int age,i,count=0;
    System.out.println("enter ur name");
    name=sc.next();
    System.out.println("enter last name");
    lname=sc.next();
    System.out.println("enter age");

    age=sc.nextInt();
    System.out.println("Age:"+age);

    System.out.println("Hello "+name);
    int len=name.length()+lname.length();
    System.out.println(len);
}
}

```

8. Write a program to input name and course of student and assign it in other class with object and function? (use two classes)

```

//import java.util.*;
class Stu
{
    String n,cou;
    java.util.Scanner sc;
    public void read()
    {
        sc=new java.util.Scanner(System.in);
        System.out.println("Enter name and course");
    }
}

```

```
n=sc.next();
cou=sc.next();
}
public void inp(String n,String course)
{
    this.n=n;
    cou=course;
}
public String name()
{
    return n;
}
public void disp()
{
    System.out.println("Hello "+n+" u r doing "+cou);
}
}
class Stu_demo
{
    public static void main(String [] sd)
    {
        Stu p=new Stu();
        //p.read();
        p.inp("swati","java");
        p.disp();
        System.out.println("bye "+p.name());
    }
}
```

9. Write a program to input number using buffered reader?

```
import java.io.*;
```

```
class Inp
{
```

```

public static void main(String [] arg)    //throws IOException
{
int n;
try{
    BufferedReader br=new BufferedReader(new
    InputStreamReader(System.in));

    System.out.println("enter a no");
    n=Integer.parseInt(br.readLine());
    System.out.println("u input :"+n);
}
catch(IOException e)
{
    System.out.println("err");
}
    System.out.println("bye..");

}
}

```

10. Write a program to input name and print length of name?

```

import java.util.*;
class inp
{
public static void main(String [] ar)
{
    Scanner sc; //reference variable
    sc=new Scanner(System.in); //object
    String name;
    String lname;

    System.out.println("enter ur name");
    name=sc.next();
}
}

```

```
        System.out.println("enter last name");
        lname=sc.next();

        System.out.println("Hello "+name);
        int len=name.length()+lname.length();
        System.out.println(len);
    }
}
```

11. Write a program to input three numbers and print their addition (using command line)?

```
class Nums
{
    public static void main(String [] aa)
    {
        int sum=0;
        int a= Integer.parseInt(aa[0]);
        int b= Integer.parseInt(aa[1]);
        int c= Integer.parseInt(aa[2]);
        sum=a+b+c;
        System.out.println("the output is "+sum);
    }
}
```

OPERATORS IN JAVA:

1. Arithmetic Operators: +, -, *, /, %
2. Unary Operators: ++, --
3. Assignment Operators : =
4. Relational Operators: <, >, <=, >=
5. Comparison operator: ==
6. Logical Operators : &&, ||, !
7. Ternary Operators : (condition) ? True statement : false statement
8. Bitwise Operators : &, |, ^
9. Shift Operators: <<, >>, >>>
10. Compound assignment: +=, -=, *=, /=, %=

11. Write a program to implement unary operator?

```
class UnaryExample{
    public static void main(String args[])
    {
        int x=10;
        System.out.println(x++);//10 (11)
        System.out.println(++x);//12
        System.out.println(x--);//12 (11)
        System.out.println(--x);//10
    }
}
```

12. Write a program to find minimum out of two numbers using ternary operator?

```
class OperatorExample
{
    public static void main(String args[])
```

```
{  
    int a=2;  
    int b=5;  
    int min=(a<b)?a:b;  
    System.out.println(min);  
}  
}
```

JAVA COMMENT:

Comments are non executable code. They can be used to provide information or explanation about the variable, method, class, or any statement.

Types of Java Comments:

There are 3 types of comments in Java:

Single Line Comment: //

Multi Line Comment: /*-----*/

Documentation Comment: /** ----- */

Control statement

1. If- else

Syntax:

```
if (expression1)
{
    // codes
}
else if(expression2)
{
    // codes
}
else if (expression3)
{
    // codes
}
.
.
else
{
    // codes
}
```

2. Switch case

Syntax:

```
switch (variable/expression)
{
    case value1:
        // statements
        break;
    case value2:
        // statements
        break;
    . . . . .
    . . . . .
    default:
        // statements
}
```

13. Write a Java program to check whether two String variable are equal

```
or not?
{
    class String_Demo
    {
        public static void main(String []
        ar) {

            String a,b="hi";
            a=ar[0];

            System.out.println("address of a:"+a.hashCode());
            System.out.println("address of b:"+b.hashCode());

            if(a==b)
                System.out.println("equal");
            else
                System.out.println("not equal");
            if(a.equals(b))
                System.out.println("equal");
        }
    }
}
```



```

else
    System.out.println("not equal");

String c=new String("sub");
String s=new String(b);
System.out.println("address of c:"+c.hashCode());
System.out.println("address of s:"+s.hashCode());

if(a==c)
    System.out.println("equal");
else
    System.out.println("not equal");
if(a.equals(c))
    System.out.println("equal");
else
    System.out.println("not equal");
    }
}

```

14. Write a program to find out maximum out of three?

```

public class JavaExample
{
    public static void main(String[] args)
    {
        int num1 = 10, num2 = 20, num3 = 7;

        if( num1 >= num2 && num1 >= num3)
            System.out.println(num1+" is the largest Number");

        else if (num2 >= num1 && num2 >= num3)
            System.out.println(num2+" is the largest Number");

        else

```

```
        System.out.println(num3+" is the largest Number");
    }
}
```

15. Write a Java program to display calculated result of numbers based on the operator entered?

```
import java.util.Scanner;

class Calculator
{
    public static void main (String[] args) {

        char operator;
        Double number1,number2,result;

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter operator (either +, -, * or /): ");
        operator = scanner.next().charAt(0);
        System.out.print("Enter number1 and number2 respectively: ");
        number1 = scanner.nextDouble();
        number2 = scanner.nextDouble();

        switch (operator) {
            case '+':
                result = number1 + number2;
                System.out.print(number1 + "+" + number2 + " = " + result);
                break;

            case '-':
                result = number1 - number2;
                System.out.print(number1 + "-" + number2 + " = " + result);
                break;

            case '*':
                result = number1 * number2;
```

```

        System.out.print(number1 + "*" + number2 + " = " + result);
        break;

    case '/':
        result = number1 / number2;
        System.out.print(number1 + "/" + number2 + " = " + result);
        break;

    default:
        System.out.println("Invalid operator!");
        break;
    }
}
}

```

16. Write a program to check whether a given year is leap year or not?

```

import java.util.Scanner;
class Calculator
{
    public static void main(String[] args)
    {
        int chk_year;
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter year");
        Chk_year=scanner.nextInt();

        if ((chk_year % 400) == 0)
            System.out.println(" year"+ chk_year+" is leap year");
        else if ((chk_year % 100) == 0)
            System.out.println(chk_year + " is a not leap year.\n");
        else if ((chk_year % 4) == 0)
            System.out.println(chk_year + " is a leap year.");
        else
            System.out.println(chk_year + " is not a leap year");
    }
}

```

17. Write a program to check whether a given number is positive / negative or zero?

```
class Ladder
{
    public static void main(String[] args)
    {

        int number = 0;

        if (number > 0)
        {
            System.out.println("Number is positive.");
        }
        else if (number < 0)
        {
            System.out.println("Number is negative.");
        }
        else
        {
            System.out.println("Number is 0.");
        }
    }
}
```

18. Write a program to calculate the amount payable after deducting the discount on the purchase amount.

Discounts are as follows:

Purchase amount $\geq 8,000$	Discount = 10% on the purchase amount
Purchase amount $\geq 5,000$	Discount = 8% on the purchase amount
Purchase amount $\geq 1,000$	Discount = 4% on the purchase amount
Otherwise	Discount = Rs. 100

```
import java.util.Scanner;
class Rates
```

```

{
    public static void main(String[] args)
    {
        double PR,dis, amount;
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter purchase rate");
        PR=scanner.nextDouble();

        If(PR>=8000)
            dis=PR*.10;
        else if(PR>=5000)
            dis=PR*.08;
        else if(PR>=1000)
            dis=PR*.04;
        else
            dis=100;

        amount=PR-dis;
        System.out.println("Amount payable after "+ dis+" discount
        is "+amount);

    }
}

```

ITERATION/ LOOPING:

A **loop** statement allows us to execute a statement or group of statements multiple times.

Java supports 3 type of loops and also enhanced for loop.

1. for loop

Java for loop provides the user to write the whole condition, i.e. initialization, condition, and increment/decrement in one line.

```
For(initialization; condition; increment/decrement){}
```

2. Enhanced for loop

This version was introduced in Java 5, this feature allows you to iterate with elements of arrays. This feature should only be used when the index is unknown.

```
for (T element:Collection obj/array)
{
    statement(s)
}
```

3. While loop

In Java while loop, a loop is continuously executed if the Boolean condition comes true, it can be considered as a repeating if statement.

```
While(condition){}
```

4. Do-while loop

A do while statement is an exit controlled loop, i.e. it checks the condition after the execution of the loop

```
do
{statements;}
while(condition);
```

19. Write a program to count vowels in given name?

```
import java.util.*;
class Vowels
{
    public static void main(String [] ar)
    {
        Scanner sc; //reference variable
        sc=new Scanner(System.in); //object
        String name;
        Int i,count=0;

        System.out.println("enter ur name");
        name=sc.next();
        System.out.println("Hello "+name);
    }
}
```

```

        for(i=0;i<name.length();i++)
        {
            char c=name.charAt(i);
            if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u')
            {
                count++;
            }
            System.out.print(c+":");
        }
        System.out.println("no. of vowels in "+name+" is "+count);
    }
}

```

20. Write a program to validate an email id: it should contain one @ and at least one "."?

```

import java.util.*;
class Scan
{
    public static void main(String [] aa)
    {
        Scanner as= new Scanner(System.in);
        String email;
        int i,count=0,count1=0;
        System.out.println(" enter the email ");
        email= as.next();
        int c=email.length();
        for(i=0;i<email.length();i++)
        {
            char k=email.charAt(i);
            if(k=='@')
            {
                count++;
            }
            if(k=='.')
            {
                count1++;
            }
        }
    }
}

```

```
        }
    }
    if(count==1&&count1>0)
    {
        System.out.println("valid id");
    }
    else
    {
        System.out.println("valid id ni h ");
    }
}
}
```

21. Write a program to reverse a given number?

```
import java.util.Scanner;
class ReverseNumberWhile
{
    public static void main(String args[])
    {
        int num=0;
        int reversenum =0;
        System.out.println("Input your number and press enter: ");
        Scanner in = new Scanner(System.in);
        num = in.nextInt();
        //While Loop: Logic to find out the reverse number
        while( num != 0 )
        {
            reversenum = reversenum * 10;
            reversenum = reversenum + num%10;
            num = num/10;
        }
        System.out.println("Reverse of input number is: "+reversenum);
    }
}
```


22. Write a program to calculate factorial of given number?

```
class FactorialDemo
{
    public static void main(String args[])
    {
        int factorial = fact(4);
        System.out.println("Factorial of 4 is: "+factorial);
    }
    static int fact(int n)
    {
        int output;
        if(n==1){
            return 1;
        }
        //Recursion: Function calling itself!!
        output = fact(n-1)* n;
        return output;
    }
}
```

23. Write a program to generate random numbers between 0 to 200?

```
import java.util.*;
class GenerateRandomNumber
{
    public static void main(String[] args)
    {
        int counter;
        Random rnum = new Random();
        /* Below code would generate 5 random numbers
        * between 0 and 200.
        */
        for (counter = 1; counter <= 5; counter++)
        {
            System.out.println(rnum.nextInt(200));
        }
    }
}
```

```
    }  
  }  
}
```

24. Write a program to print fibonacci series 0,1,1,2,3....N ?

```
import java.util.Scanner;  
class ReverseNumberWhile  
{  
    public static void main(String args[])  
    {  
        int num=0;  
        int i=0,j=1;  
        System.out.println("Input your number and press enter: ");  
        Scanner in = new Scanner(System.in);  
        num = in.nextInt();  
        while(i<=num)  
        {  
            System.out.println(i);  
            j=i+j;  
            i=j-i;  
        }  
    }  
}
```

25. Write a program to find first N prime numbers?

```
import java.util.Scanner;  
  
class PrimeNumberDemo  
{  
    public static void main(String args[])  
    {  
        int n;  
        int status = 1;  
        int num = 3;
```

```
//For capturing the value of n
Scanner scanner = new Scanner(System.in);
System.out.println("Enter the value of n:");
//The entered value is stored in the var n
n = scanner.nextInt();
if (n >= 1)
{
    System.out.println("First "+n+" prime numbers are:");
    //2 is a known prime number
    System.out.println(2);
}

for ( int i = 2 ; i <=n ; )
{
    for ( int j = 2 ; j <= Math.sqrt(num) ; j++ )
    {
        if ( num%j == 0 )
        {
            status = 0;
            break;
        }
    }
    if ( status != 0 )
    {
        System.out.println(num);
        i++;
    }
    status = 1;
    num++;
}
}
```

26. Write a program to input Login details:user name and password, password should not be visible at runtime?

```
import java.util.*;
import java.io.*;
class Login
{
    public static void main(String [] args)
    {
        String uname,pwd;
        char [] p=new char[10];
        Scanner sc=new Scanner(System.in);
        Console con=System.console();
        System.out.println("Enter user name");
        uname=sc.next();
        //uname=con.readLine();
        System.out.println("Enter user password");
        //pwd=sc.next();
        p=con.readPassword();
        System.out.println("Hello user :"+uname);
        //System.out.println("U entered Password :"+pwd);
    }
}
```

For pattern and loops example please refer “C programming and coding question bank with solution”.

ARRAY:

Array is an object that contain elements of the similar data type. We can store only fixed set of elements in a Java array.

Once defined, the size of an array is fixed and cannot increase to accommodate more elements. The first element of an array starts with index zero.

Syntax:

```
Datatype[] arrname;  
arrname=new datatype[size of array];  
datatype [][] arrname=new datatype[rows][cols];
```

Advantages of array:

- Code Optimization: It makes the code optimized, we can retrieve or sort the data easily.
- Random access: We can get any data located at any index position.

27. Write a program to input five numbers in an array and calculate their sum and find maximum?

```
import java.util.*;  
class array1  
{  
    public static void main(String []at)  
    {  
        Scanner sc=new Scanner(System.in);  
        int []a=new int[5];  
        int i,s=0,max=0;  
        for(i=0;i<5;i++)  
        {  
            System.out.println("Enter no.");  
            a[i]=sc.nextInt();  
            s=s+a[i];  
            if(a[i]>max)
```

```
        max=a[i];
    }
    for(i=0;i<5;i++)
    {
        System.out.println("a[" + i + "]= " + a[i]);
    }
    System.out.println("Sum =" +s);
    System.out.println("Max="+max);
}
}
```

28. Write a program to print different number of columns in a 2-D array?

```
import java.util.*;
class Array3
{
    public static void main(String []at)
    {
        Scanner sc=new Scanner(System.in);
        int a[][]=new int[][] {
            {1,2} ,
            {2,3,4} ,
            {5,6,9} ,
            {8}
        };
        int i,j;

        for(i=0;i<a.length;i++)
        {
            for(j=0;j<a[i].length;j++)
            {
                //System.out.print("a[" + i + "][" +j+"]= " + a[i][j]);
                System.out.print(a[i][j]+"\\t");
            }
        }
    }
}
```

```

        System.out.println();
    }
}
}

```

29. Write a program to initialize String array and print it?

```

import java.util.*;
class array4
{
    public static void main(String []at)
    {
        Scanner sc=new Scanner(System.in);
        String [] col=new String[]{"red","green","blue"};
        int i;
        for(i=0;i<col.length;i++)
        {
            System.out.println(col[i]+": "+col[i].length());
        }
    }
}

```

30. Write a program to print random captcha?

```

import java.util.*;
class sgn
{
    public static void main(String [] aa)
    {
        Random r= new Random();

        char [] q1=new char[] {'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};
        char [] q2=new char[] {'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};
        int f = r.nextInt(20);
    }
}

```

```
        System.out.print(q2[f]);
        System.out.print(f+100*f+q1[f]);

        System.out.print(q1[f/2]);
    }
}
```

31. Write a program to demonstrate three-dimensional array?

```
class ThreeD
{
    public static void main(string [] arg)
    {
        int threed [ ] [ ] [ ] ==new int [3][4][5];
        int i,j,k;
        for(i=0;i<3;i++)
            for(j=0;j<4;j++)
                for(k=0;k<5;k++)
                    threed[i][j][k]= i * j * k;

        for(i=0;i<3;i++) {
            for(j=0;j<4;j++) {
                for(k=0;k<5;k++)
                    System.out.print(threed[i][j][k] + " ");
                System.out.println();
            }
        }
        System.out.println();
    }
}
```

32. Write a program to demonstrate labeled break?

```
class Break
{
    public static void main(String [] arg)
```



```

{
    boolean flag=true;

    first:{
        second:{
            third:{
                System.out.println("Before break");
                If(flag) break second; //break out of second block
                System.out.println("this wont' execute");
            }
            System.out.println("this wont' execute");
        }
        System.out.println("this is after second block");
    }
}
}
}
}

```

33. Write a program to print triangular multiplication table for 0 to 9?

```

class Tables
{
    public static void main(String [] arg)
    {
        outer: for(int i=0;i<10;i++){
            for(int j=0;j<10;j++){
                if(j>i){
                    System.out.println();
                    Continue outer;
                }
                System.out.println(" "+( i * j));
            }
        }
        System.out.println();
    }
}

```

OBJECT AND CLASSES

Object is a real world entity which has some identity, state, behaviour.

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

A class can have the following:

- fields
- methods
- constructors
- blocks
- nested class and interface

A variable which is created inside the class but outside the method, is known as instance variable.

Instance variable doesn't get memory at compile time. It gets memory at run time when object (instance) is created. That is why, it is known as instance variable.

The **new** keyword is used to allocate memory at run time. All objects get memory in Heap memory area.

First letter of class name in java should be capital (it is a rule of java) but if you write it in small there is no error.

34. Write a program to create different classes for student and employee with methods and fill details?

```
import java.util.*;
class emp
{
    String job;
    Scanner sc;
    public void inp()
    {
        sc=new Scanner(System.in);
```

```
        System.out.println("enter post");
        job=sc.next();
    }
    public void put()
    {
        System.out.println("u r working as "+job);
    }
}
class stu
{
    String name;
    int age;
    Scanner sc;
    public void get()
    {
        sc=new Scanner(System.in);
        System.out.println("Enter name and age");
        name=sc.next();
        age=sc.nextInt();
    }
    public void put()
    {
        System.out.println("hello "+name);
        System.out.println("u r "+age+" years old");
    }
}
class stu_demo
{
    public static void main(String [] ar)
    {
        stu s=new stu();
        emp e=new emp();
        System.out.println("Enter stu data");
        s.get();
        System.out.println("Enter employee record");
```

```
e.inp();
System.out.println("Data Input:");
s.put();
e.put();
}
}
```

35. Write a program to demonstrate implicit and explicit conversion?

```
class Explicit
{
    public static void main(String [] ar)
    {
        Object o,o1;
        String s = new String("Hello");
        o = new String("Hello");
        String t;
        t = (String) o;
        o1=t;//implicit
    }
}
```

36. Write a program to check whether a given string is palindrome or not?

```
import java.util.*;
class count_word
{
    public static void main(String [] as)
    {
        String n;
        int i,wc=1,cc=0,space=0,j;
        char c,c1;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a string");
        n=sc.nextLine();
        for(i=0;i<n.length();i++)
```

```
{
    c=n.charAt(i);

    if(c==' ')
    {
        space++;
        wc++;
    }
}
cc=n.length()-space;
System.out.println("Total characters in name are" + cc + "\n
Total words are " + wc);

//Palindrome string

space=0;
for(i=0,j=n.length()-1;i<n.length()/2;i++,j--)
{
    c=n.charAt(i);
    c1=n.charAt(j);
    if(c!=c1)
    {
        space=1;
        break;
    }
}
if(space==0)
System.out.println("Palindrome string");
else
System.out.println("Not");
}
}
```

37. Write a program to call methods of a class using anonymous object?

```
class Factorial
{
    void fact(int n)
    {
        int fact=1;
        for(int i=1;i<=n;i++)
        {
            fact=fact*i;
        }
        System.out.println("factorial is "+fact);
    }
    public static void main(String args[]){
        new Factorial().fact(5);//calling method with anonymous object
    }
}
```

38. Write a program to calculate the area of rectangle, create method get() to input values and calculateArea() to calculate area?

```
class Rectangle
{
    int length;
    int width;
    void get(int l,int w)
    {
        length=l;
        width=w;
    }
    void calculateArea()
    {
        System.out.println(length*width);}
}
class TestRectangle2
{
```

```
public static void main(String args[]){
    Rectangle r1=new Rectangle(),r2=new Rectangle();//
    creating two objects
    r1.get(11,5);
    r2.get(3,15);
    r1.calculateArea();
    r2.calculateArea();
}
}
```

39. Write a program to create a class Account with different methods like deposit(), withdraw() and operate?

```
class Account
{
    int acc_no;
    String name;
    float amount;
    void open_account(int a,String n,float amt)
    {
        acc_no=a;
        name=n;
        amount=amt;
    }
    void deposit(float amt)
    {
        amount=amount+amt;
        System.out.println(amt+" deposited");
    }
    void withdraw(float amt)
    {
        if(amount<amt)
        {
            System.out.println("Insufficient Balance");
        }
        else{
```

```
        amount=amount-amt;
        System.out.println(amt+" withdrawn");
    }
}

void checkBalance()
{System.out.println("Balance is: "+amount);}
void display()
{System.out.println(acc_no+" "+name+" "+amount);}
}

class TestAccount{
public static void main(String[] args){
Account a1=new Account();
a1.open_account (212345,"Shweta",10000);
a1.display();
a1.checkBalance();
a1.deposit(40000);
a1.checkBalance();
a1.withdraw(15000);
a1.checkBalance();
}}
```

40. Write a program to create set() and get() methods to read/ write values of student data?

```
class Student
{
    int roll_no;
    String course;
    void set_roll(int r)
    {
        Roll_no=r;
    }
    int get_roll()
    {
        return (roll_no);
    }
}
```



```
    }  
    void set_course(String c)  
    {  
        course=c;  
    }  
    int get_course()  
    {  
        return(course);  
    }  
    public static void main(String [] arg)  
    {  
        Student s=new Student();  
        s.set_roll(101);  
        s.set_course("java");  
        System.out.println("Roll no"+s.get_roll()+” doing “+s.get_course);  
    }  
}
```

CONSTRUCTOR:

Constructor is a member function of a class. It is called automatically when an instance of an object is created and memory is allocated for the object.

It is used to initialize an object.

Every class has at least one constructor called default constructor.

Keypoints:

- Constructor name is same as class name.
- It has no return type.
- It is called automatically we need not to call explicitly.
- Its overloading is possible means a class can have any number of constructor with different parameters.
- It can be of two type: default , parameterized

41. Write a program to demonstrate default, parameterized and copy constructor?

```
class Constructor_demo
{
    int x;
    Constructor_demo ()
    {
        System.out.println("default constructor");
    }
    Constructor_demo (int a)
    {
        this.x=a;
        System.out.println("parameterized constructor " + a);
    }
    Constructor_demo (Constructor_demo a)
```

```

    {
        System.out.println("copy constructor" +a.x);
    }
    public static void main(String [] sd)
    {
        Constructor_demo d=new Constructor_demo ();
        d=new Constructor_demo (1);
        Constructor_demo d1=new Constructor_demo (d);
    }
}

```

42. Write a program to use **variable number of argument** for addition any number of values?

```

class Addi
{
    int t;
    public void sum(int ...a)
    {
        t=0;
        for(int i:a)
        {
            t=t+i;
        }
        System.out.println("Sum="+t);
    }
    public static void main(String [] arg)
    {
        Addi a=new Addi();
        a.sum(1,3,4,6,7,5,4,3,3,2,3,4,4);
        a.sum(45,23);
    }
}

```

43. Write a program using constructors input data of student?

```
import java.util.*;
class student1
{
    String name,course;
    student1()
    {
        name="guest";
        course="not decided";
    }
    student1(String name)
    {
        this.name=name;
        course="not decided";
    }
    student1(String name,String course)
    {
        this.name=name;
        this.course=course;
    }
    void read(String name, String c)
    {
        this.name=name;
        System.out.println("reference of this"+this.hashCode());
        course=c;
    }
    void show()
    {
        System.out.println("Hello "+name);
        System.out.println("u r doing"+course);
    }
}
class Student
{
```

```

public static void main(String [] arg)
{
    student1 s=new student1();
    System.out.println("reference of s"+s.hashCode());
    String a,b;
    Scanner sc=new Scanner(System.in);
    /*System.out.println("Enter naem and course");
    a=sc.next();
    b=sc.next();*/
    //s.read(a,b);
    s.show();
    s=new student1("surabhi");
    System.out.println("reference of s"+s.hashCode());
    s.show();
    student1 s1=new student1("raj", "java");
    System.out.println("reference of s1"+s1.hashCode());
    s1.show();
}
}

```

44. Write a program to call constructor within another constructor?
(constructor chaining)

```

public class Chaining
{
    public Chaining()
    {
        System.out.println("In default constructor...");
    }
    public Chaining(int i)
    {
        this();
        System.out.println("In single parameter constructor...");
    }
}

```

```
    }  
    public Chaining(int i,int j){  
        this(j);  
        System.out.println("In double parameter constructor...");  
    }  
  
    public static void main(String a[]){  
        Chaining ch = new Chaining(10,20);  
    }  
}
```

STATIC:

The static keyword in Java is used for memory management mainly.

The static can be:

1. variable (also known as class variable)
2. method (also known as class method)
3. block
4. nested class

A static variable is common to all the instances (or objects) of the class because it is a class level variable.

In other words you can say that only a single copy of static variable is created and shared among all the instances of the class.

Memory allocation for such variables only happens once when the class is loaded in the memory.

45. Write a program to assign same college name to students on ID card of different students. Student Name and branch should ask ?

```
class Student_ID_Card
{
    String name, branch;

    static String college ="Swati Computers";

    Student_ID_Card (String n,String b)
    {
        name = n;
        branch=b;
    }
    void display (){System.out.println(name+" " + branch + " "+college);
}

public static void main(String args[])
{
```

```
Student_ID_Card s1 = new Student_ID_Card ("Sweety","CS");
Student_ID_Card s2 = new Student_ID_Card ("Shweta","ECE");
```

```
s1.display();
s2.display();
}
}
```

46. Write a program to demonstrate static block?

```
class static_demo
{
    static int a;
    static
    {
        System.out.println("hello");
    }
    static
    {
        System.out.println("wel");
    }
    public static void main(String [] ar)
    {

        a=10;
        System.out.println("hi");
    }
}
```

47. Write a program to count the number of students those who have taken admission in different branches of a college?

```
class static_demo1
{
    static int count=0;
    static_demo1(String b)
```



```

    {
        System.out.println(b);
        count++;
    }

    public static void put()
    {
        System.out.println(count+"");
    }
    public static void main(String [] ra)
    {
        static_demo1 s=new static_demo1("CS");
        static_demo1 s1=new static_demo1("ECE");
        static_demo1 s2=new static_demo1("Electrical");
        static_demo1.put();
    }
}

```

48. Write a program to demonstrate how to access a variable of another class without object?

```

class Static_demo
{
    static int a=100;
}

class Static_Demo
{
    public static void main(String [] ar)
    {
        System.out.println(" a = " + Static_demo.a);
    }
}

```

49. Write a program to differentiate between instance variable and class variable?

```
class Stu1
{
    int count; //instance variable
    static int cou; //class variable
    public Stu1()
    {
        count++;
        cou++;
    }
    public void display()
    {
        System.out.println(count);
    }
}
class Third
{
    static{
        System.out.println("hi");
    }
    static
    {
        System.out.println("hii");
    }
    public static void main(String [] ar)
    {
        Stu1 s=new Stu1();
        Stu1 s1=new Stu1();
        s.display();
        s1.display();
        System.out.println("Total students"+Stu1.cou);
    }
}
```

“THIS” KEYWORD:

This is a **reference variable** that refers to the current object.

Usage of Java's *this* keyword.

- \$ this can be used to refer current class instance variable.
- \$ this can be passed as an argument in the method call.
- \$ this() can be used to invoke current class constructor.
- \$ this can be passed as argument in the constructor call.
- \$ this can be used to invoke current class method (implicitly)
- \$ this can be used to return the current class instance from the method.

50. Write a program to call methods of a class from another method of same class?

```
class Methods
{
    void two()
    {
        System.out.println("hello two");}
    void one()
    {
        System.out.println("hello one");
        this.two();
    }
}
class Test_This
{
    public static void main(String args[]){
        Methods m=new Methods();
        m.one();
    }
}
```

51. Write a program to input name and roll number of students?

```
class Students
{
    int roll;
    String name;
    public void get(int r,String name)
    {
        Roll=r;
        this.name=name;
    }
    public void put()
    {
        System.out.println("roll no="+roll+" name="+name);
    }
    public static void main(String [] ae)
    {
        Students s=new Students();
        s.get(1,"shweta");
        s.put();
    }
}
```

52. Write a program to call constructor of same class in another constructor?

```
class Constr
{
    Constr()
    {System.out.println("I am a default constructor,called by constructor");}
    Constr (int x)
    {
        this();
        System.out.println(x);
    }
}
class Test_This
```

```

    {
        public static void main(String args[]){
            Constr c=new Constr(10);
        }
    }

```

53. Write a program to call parameterized constructor by default constructor of same class?

```

class A
{
    A(){
        this(10);
        System.out.println("I am a default constructor");
    }
    A(int x)
    {
        System.out.println("I am a parameterized constructor");
    }
}
class Test_This{
    public static void main(String [] arg){
        A a=new A();
    }
}

```

54. Write a program to pass "this" as an argument?

```

class This_test
{
    void fun(This_test obj)
    {
        System.out.println("method is invoked");
    }
    void funct()
    {
        fun(this);
    }
}

```

```
    }  
    public static void main(String args[]){  
        This_test s1 = new This_test ();  
        s1.funct();  
    }  
}
```

Final Keyword:

Final keyword is use to make constant and restrict user to use class, variable, or methods.

Final can be:

- Variable: you cannot change the value of final variable (It will be constant).
- Class: If you make any class as final, you cannot extend it.
- Method: If you make any method as final, you cannot override it.

55. Write a program to give name of college, student name, year in which he/she is studying on ID card. Name of college cannot change for any student?

```
Class Id_card  
{  
    final String name_college="Swati Computers";  
    String name_stu;  
    int year;  
    public Id_card()  
    {  
        name_stu="Unknown";  
        year=0;  
    }  
    public Id_card(String n,int y)  
    {  
        name_stu=n;  
        year=y;  
    }  
}
```

```

    }
    public void display()
    {
        System.out.println(name_stu +” student of “+ name_college+”
        is studying in “+ year +” year”);
    }
    public static void main(String [] ar)
    {
        Id_card i=new Id_card();
        Id_card i1=new Id_card(“swati”,1);
        Id_card i2=new Id_card(“shweta”,2);

        i.display();
        i1.display();
        i2.display();
    }
}

```

56. Write a program to demonstrate a problem : If you have your fashion designing institute name “SWFT” and you do not want to give any franchise of it. If any other institute displays on board that it was a concern of “SWFT”, it is illegal. How to solve this in Java class?

```
final class SWFT{
```

```

    class Any_institute extends SWFT
    {
        public Any_institute()
        {
            System.out.println(“We are sister concern of SWFT”);
        }
        public static void main(String [] at)
        {
            new Any_institute();
        }
    }
}

```

57. As SBI bank has many branches in india and any of the customer can use its service from any branch but if any customer is facing any problem regarding cheque book/ATM etc, he/she should consult to the main branch as the Manager said. Main branch have many rights, but small branches don't. Write Java classes to create main and small branches of SBI and some of the functionality cannot be inherited in small branches.

```
class SBI_main
{
    SBI_main()
    {
        System.out.println("If you have issues related cheque book
etc,yours welcome");
    }

    final void cheque_book_issue(){}
    final void amount_issue(){}

    void withdraw(){}
    void deposit(){}
    void open_new_account(){}
}

class SBI_Mansarovar extends SBI_main
{
    void withdraw(){}
    void deposit(){}
    void open_new_account(){}
}

class SBI_Tnokroad extends SBI_main
{
    void withdraw(){}
    void deposit(){}
    void open_new_account(){}
}
```


JAVA - REGULAR EXPRESSIONS (REGEX)

The Java Regex or Regular Expression is an API to define pattern for searching or manipulating strings.

A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern.

Java.util.regex package contains 1 interface and 3 classes:

1. MatchResult interface
2. Matcher class
3. Pattern class
4. PatternSyntaxException class

58. Write a Java program to validate email using regeX?

```
import java.util.regex.*;
public class Validate_email {
    public static void main(String[] args) {
        String EMAIL_REGEX = "^[\\w-\\.]*[\\w-\\.]\\@([\\w]+\\.\\.)+[\\w]+[\\w]$";
        String email1 = "user@domain.com";
        Boolean b = email1.matches(EMAIL_REGEX);
        System.out.println("is e-mail: "+email1+":Valid = " + b);
        String email2 = "user^domain.co.in";
        b = email2.matches(EMAIL_REGEX);
        System.out.println("is e-mail: "+email2+":Valid = " + b);
    }
}
```

59. Write a program to search the occurrence of word “book” in a given string using regex?

```
import java.util.regex.*;
class RegexExample1{
    public static void main(String args[]){
        String content = “This is a java “ +
            “book from swati”;

        String pattern = “.*book.*”;

        boolean isMatch = Pattern.matches(pattern, content);
        System.out.println(“The text contains ‘book’? “ + isMatch);
    }
}
```

60. Write a program to replace all occurrence of word “tough” by “easy” in a given string?

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class RegexMatches {

    private static String REGEX = “tough”;
    private static String INPUT = “All says Java is a tough language “ +
        “Java programs are tough”;
    private static String REPLACE = “easy”;

    public static void main(String[] args) {
        Pattern p = Pattern.compile(REGEX);

        // get a matcher object
        Matcher m = p.matcher(INPUT);
        INPUT = m.replaceAll(REPLACE);
        System.out.println(INPUT);
    }
}
```

STRING

Generally, string is a sequence of characters. But in Java, string is an object that represents a sequence of characters. The `java.lang.String` class is used to create string object.

Java String literal is created by using double quotes.

The Java String is immutable i.e. it cannot be changed. Whenever we change any string, a new instance is created.

For mutable string, you can use `StringBuffer` and `StringBuilder` classes.

61. Write a program to replace 'a' by 'E' in given String?

```
import java.io.*;
public class Test {

    public static void main(String args[]) {
        String Str = new String("java language");

        System.out.print("Return Value :");
        System.out.println(Str.replace('a', 'E'));

    }
}
```

Difference between `StringBuffer` and `StringBuilder`

StringBuffer

`StringBuffer` is synchronized i.e. thread safe. It means two threads can't call the methods of `StringBuffer` simultaneously.

StringBuilder

`StringBuilder` is *non-synchronized* i.e. not thread safe. It means two threads can call the methods of `StringBuilder` simultaneously.

62. Write a Java program to demonstrate difference between String and String Builder class?

```
public class strng {  
    public static void main(String [] ar)  
    {  
        String n=new String("Swati");  
        System.out.println(n.hashCode());  
        n="saxena";  
        System.out.println(n.hashCode());  
        n=n+"cloud";  
        System.out.println(n.hashCode());  
        StringBuilder sb=new StringBuilder("Swati");  
        System.out.println(sb.hashCode());  
        sb.append("saxena");  
        System.out.println(sb);  
        System.out.println(sb.hashCode());  
    }  
}
```

63. Write a Java program to explain the use of startWith() of String ?

```
public class strng {  
    public static void main(String [] ar)  
    {  
        String str= new String("swati computers, developers choice");  
        System.out.println("String str starts with swati: "+str.  
            startsWith("swati"));  
        System.out.println("String str starts with brown: "+str.  
            startsWith("brown"));  
        System.out.println("substring of str(starting from 6th index) has  
            computers prefix: "  
            +str.startsWith("computers", 6));  
        System.out.println("substring of str(starting from 17th index)  
            has developers prefix: "  
            +str.startsWith("developers", 17));  
    }  
}
```

64. Write a program to count 'a' in a given string?

```
class A
{
public static void main(String [] ar)
{
String n=ar[0];
int i,count=0;
for(int i=0;i<n.length();i++)
{
    if(n.charAt(i)=='a')
    {count++; }
}
System.out.println("total a in string are:"+count);
}
}
```

INSTANCEOF

The Java instanceof operator is used to test whether the object is an instance of the specified type (class or subclass or interface).

65. Write a program to check that the given object is instance of which class?

```
class Person
{
    public void show()
    {
        System.out.println("Person");
    }
}
class stu extends Person
{
    public void show()
    {
        System.out.println("stu");
    }
}
class emp extends Person
{
    public void show()
    {
        System.out.println("emp");
    }
}
public class Instance
{
    public static void fun(Person p)
    {
        if(p instanceof stu)
```

```
        p.show();
    else
        p.show();
}
public static void main(String [] arg)
{
    Person p;
    p=new stu();
    Instance.fun(p);
    p=new emp();
    Instance.fun(p);
}
}
```

INNER CLASS

Inner class is a class which is declared inside another class. Java allowed this. Class that holds the inner class is called outer class.

There are basically four types of inner classes in Java.

- 1) Nested Inner class
- 2) Method Local inner classes
- 3) Anonymous inner classes
- 4) Static nested classes

Inner class can access private members of outer class.

66. Write a program to create an inner class and call its method?

```
class outer
{
    String n;

    public void get()
    {
        class inner
        {
            public void show()
            {
                System.out.println("hi");
            }
        }
        n="outer";
        System.out.println(n);
        new inner().show();
    }
}

public class Inners {
    public static void main(String [] rag)
```



```

    {
        outer o=new outer();
        o.get();
    }
}

```

67. Write a program to demonstrate method local inner class?

```

class Outer {
    void outerMethod() {
        System.out.println("inside outerMethod");
        class Inner {
            void innerMethod() {
                System.out.println("inside innerMethod");
            }
        }
        Inner y = new Inner();
        y.innerMethod();
    }
}

class Method_Inner_Demo {
    public static void main(String[] args) {
        Outer x = new Outer();
        x.outerMethod();
    }
}

```

68. In a large project you may have classes with common name, Give an idea how to resolve this naming clash?

```

class OuterStatic
{
    private int mem = 20;
    private static int smem = 50;
}

```

```
static class InnerStatic
{
    public void accessMembers ()
    {
        System.out.println(mem); //Error: Cannot make a static
        reference to the non-
        //static field mem
        System.out.println(smeme);
    }
}

public class StaticClassDemo
{
    public static void main(String[] args)
    {
        OuterStatic.InnerStatic is = new OuterStatic.InnerStatic();
        is.accessMembers();
    }
}
```

INHERITANCE:

Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another.

`extends` is the keyword used to inherit the properties of a class.

Super

The `super` keyword

It is used to differentiate the members of superclass from the members of subclass, if they have same names.

It is used to invoke the superclass constructor from subclass.

Overriding

If a class inherits a method from its super class, then there is a chance to override the method provided that it is not marked `final`.

The benefit of overriding is: ability to define a behaviour that's specific to the subclass type which means a subclass can implement a parent class method based on its requirement.

Rules for method overriding:

- The argument list should be exactly the same as that of the overridden method.
- The return type should be the same or a subtype of the return type declared in the original overridden method in the superclass.
- The access level cannot be more restrictive than the overridden method's access level. For example: if the superclass method is declared `public` then the overriding method in the sub class cannot be either `private` or `protected`.
- Instance methods can be overridden only if they are inherited by the subclass.

- A method declared as final cannot be overridden.
- A method declared as static cannot be overridden but can be re-declared.
- If a method cannot be inherited, then it cannot be overridden.
- A subclass within the same package as the instance's superclass can override any superclass method that is not declared private or final.
- A subclass in a different package can only override the non-final methods declared public or protected.
- An overriding method can throw any unchecked exceptions, regardless of whether the overridden method throws exceptions or not.
- However the overriding method should not throw checked exceptions that are new or broader than the ones declared by the overridden method.
- The overriding method can throw narrower or fewer exceptions than the overridden method.
- Constructors cannot be overridden.

69. Write a program :create a Base class with default and parameterized constructor and a derived class with default and parameterized constructor and create objects of derived class?

```
class base1
{
    public base1()
    {
        System.out.println("base default constructor called");
    }
    public base1(String t)
    {
        System.out.println("parameterized constructor base:"+t);
    }
}
class sub1 extends base1
{
    public sub1()
```

```

    {
        System.out.println("sub default constructor called");
    }
    public sub1(String t)
    {
        super(t);
        System.out.println("parameterized constructor derived:"+t);
    }

}
class inheri_demo1
{
    public static void main(String [] ra)
    {
        sub1 s=new sub1();
        sub1 s1=new sub1("hi");
    }
}

```

70. Write a Java class "student" which inherits "person" class . person class have members name and age, and student class having course as member?

```

class Person
{
    String name;
    int age;
    public void read(String n,int age)
    {
        name=n;
        this.age=age;
    }
    public void show()
    {
        System.out.println("Hello "+name);
        System.out.println("Age is"+age);
    }
}

```

```
    }  
    class student extends Person  
    {  
        String course;  
        public void get(String n,int a,String co)  
        {  
            read(n,a);  
            course=co;  
        }  
        public void show()  
        {  
            super.show();  
            System.out.println("course"+course);  
        }  
    }  
    class inheri1  
    {  
        public static void main(String [] arg)  
        {  
            student s=new student();  
            s.get("rahul",22,"java");  
            s.show();  
        }  
    }  
}
```

71. Create a Java class "teacher" with properties like designation (assistant, associate professor), name of college . create class "physics", chemistry" which inherits "teacher" class and its properties.

```
class Teacher  
{  
    String designation = "Teacher";  
    String collegeName = "SwatiComputers";  
    public String getDesignation()  
    {  
        return designation;  
    }  
}
```

```
    }  
    protected void setDesignation(String designation)  
    {  
        this.designation = designation;  
    }  
    protected String getCollegeName()  
    {  
        return collegeName;  
    }  
    protected void setCollegeName(String collegeName)  
    {  
        this.collegeName = collegeName;  
    }  
  
    void does()  
    {  
        System.out.println("Teaching");  
    }  
}
```

```
class PhysicsTeacher extends Teacher  
{  
class ChemistryTeacher extends Teacher  
{  
Class teacher_demo{  
    public static void main(String args[]){  
        PhysicsTeacher obj = new PhysicsTeacher();  
        obj.does();  
        ChemistryTeacher obj1=new ChemistryTeacher();  
        Obj1.does();  
    }  
}
```

ABSTRACTION

Abstraction is a process of hiding the implementation details and showing only functionality to the user.

In other words it is the process to show the features which are necessary for current problem scenario.

For example, if you want to switch on the fan, you will click on the switch but you don't know the internal processing about the fan.

There are two ways to achieve abstraction in Java

1. Abstract class (0 to 100%)
2. Interface (100%)

Abstract class

A class which contains implemented and non implemented methods, declared as abstract and called abstract class.

- Abstract class cannot be instantiated.
- Abstract classes may or may not contain abstract methods, i.e., methods without body (`public void fun();`)
- To use an abstract class, you have to inherit it from another class, provide implementations to the abstract methods in it.
- If you inherit an abstract class, you have to provide implementations to all the abstract methods in it otherwise derived class should be declare as abstract.

72. Write a Java class to explain abstraction?

```
abstract class abst
{
    public void x()
    {
        System.out.println("x() in abstract");
    }
    abstract public void y();
}
```



```

class abst_demo extends abst
{
    public void y()
        {System.out.println("y()");}

    public static void main(String [] ra)
    {
        //abst ab=new abst();
        abst_demo a=new abst_demo();
        a.x();
        a.y();
    }
}

```

73. Write an Base abstract class, create two classes which inherits Base class, in another class create array of Base class and give reference of derived classes and call methods?

```

abstract class Base
{
    public Base()
    {
        System.out.println("base() called");
    }
    abstract public void call();
}
class Base1 extends Base
{
    public void call()
    {
        System.out.println("a() called");
    }
}
class Base2 extends Base
{
    public void call()

```

```
        {
            System.out.println("b() called");
        }
    }
}
class instance_demo
{
    public static void main(String[] sg)
    {
        Base1 b1=new Base1();
        Base2 b2=new Base2();
        Base[] b=new Base[2];
        b[0]=b1;
        b[1]=b2;
        for(int i=0;i<2;i++)
        {
            if(b[i] instanceof Base1) // to check which derived class
                object's reference is in abstract class's
            {
                //reference variable instance of is used
                ((Base1)b[i]).call(); //typing casting because b is a
                reference variable of abstract class
            }
            else
            {
                ((Base2)b[i]).call();
            }
        }
    }
}
```

Interface

An interface is a reference type in Java. It has static constants and abstract methods.

A class can implement more than one interface but can extend single class.

It is used to achieve abstraction and multiple inheritance in Java. It represents IS-A relationship.

74. Create two interface and a class to implement interfaces?

```
interface intr
{
    public void put();
}
interface intr1
{
    final String name="SwatiComputers";
    public void display();
    public void get();
}

class intr_demo implements intr,intr1
{
    public void fun(){System.out.println("hello");}
    public void get(){}
    public void put()
    {
        //System.out.println(name);
        System.out.println("put() defined");
    }
    public static void main(String [] arg)
    {
        new intr_demo().put();
    }
}
```

75. Create two interface and one interface which extends another interface and a class which implement these interface?

```
interface intr1
{
    public void x();
```

```
        public void y();
    }
    interface intr2
    {
        public void z();
    }
    interface intr3 extends intr2 //An interface can inherit another interface
    {
        public void a();
    }
    class intr_demo implements intr1,intr3 //A class can implement more
than 1 interface
    {
        public void x()
        {
            System.out.println("x() of interface");
        }
        public void a()
        {
            System.out.println("a() called");
        }
        public void z()
        {
            System.out.println("z() called");
        }
        public void y(){}
        public static void main(String [] ar)
        {
            intr_demo d=new intr_demo();
            intr3 i=d;
            d.x();
            i.a();
            i.z();
        }
    }
```

EXCEPTION

Exceptions are the runtime errors which can be handle by exception handling mechanism to avoid abnormal termination of program.

Type of Exception

1. Checked
2. Unchecked

Exception Handling Keywords:

try
catch
finally
throw
throws

76. Write a Java class to input name using BufferedReader?

```
import java.util.*;  
import java.io.*;  
class exc  
{  
    public static void main(String [] ar)throws Exception  
    {  
        BufferedReader br=new BufferedReader(new  
        InputStreamReader(System.in));  
        System.out.println("Enter name");  
        String name=br.readLine();  
    }  
}
```

77. Write a program to input a number and a string using command line argument and handle `ArrayIndexOutOfBoundsException`, `NumberFormatException`?

```
class exc1
{
    public static void main(String [] ar)
    {
        int w=6,r;
        try{
            int q=Integer.parseInt(ar[0]);
            System.out.println("q="+q);
            String n=ar[1];
            System.out.println(" n="+n);
            r=w/q;
            System.out.println("r="+r);
        }
        catch(NumberFormatException ex)
        {
            System.out.println("enter numeric value");
        }
        catch(ArrayIndexOutOfBoundsException ex)
        {
            System.out.println("enter 2 arguments");
        }
        catch(Exception ex)
        {
            System.out.println(ex.toString());
        }
        Finally
        {
            System.out.println("i m ready to execute");
        }
    }
}
```

78. Write a Java program to enter two numbers, if the number is 0, handle NumberDivided By Zero Exception (ArithmeticException)?

```
import java.util.*;
class Excep1
{
    public static void main(String [] ar)
    {
        double c;
        int a,b;
        Scanner sc;
        sc=new Scanner(System.in);
        System.out.println("Enter 2 nos");
        try{a=sc.nextInt();
            b=sc.nextInt();
            c=a/b;
            System.out.println("Answer="+c);
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println(e.getMessage());
        }
        catch(Exception e)
        {
            System.out.println(e.getMessage());
        }
        finally
        {
            System.out.println("Finished");
        }
    }
}
```

79. Write a user defined exception to validate if age is below 18 then teenage exception should throw?

```
class myexp extends Exception
{
    public myexp(String er)
    {
        super(er);
    }
}
class vali_age
{
    public void valid(int ag)
    {
        try{    if(ag<18)
            throw new myexp("teenager");
            else
                System.out.println("Hi");
        }
    }
    catch(myexp e)
    {
        System.out.println(e.getMessage());
    }
}
class validate
{
    public static void main(String [] ar)
    {
        try{
            vali_age v=new vali_age();
            v.valid(20);
            v.valid(12);
        }
        catch(Exception e)
        {
```



```

        System.out.println(e.getMessage());
    }
}
}

```

80. Write a program to handle InputMismatchException?

```

import java.util.*;
class exc
{
    public static void main(String [] ar)
    {
        int age;
        Scanner sc=new Scanner(System.in);
        try
        {
            System.out.println("enter ur age");
            age=sc.nextInt();
            System.out.println("age is "+age);
        }
        catch(InputMismatchException ex)
        {
            System.out.println("Enter age in numbers..example:23");
        }

        System.out.println("thanks for using program");
    }
}

```

The try-with-resources statement

In Java, the try-with-resources statement is a try statement that declares one or more resources.

The resource is as an object that must be closed after finishing the program.

The try-with-resources statement ensures that each resource is closed at the end of the statement execution.

81. Write a Java program to open multiple resource using try with resource, code to read and write data into file?

```
import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;

public class TryWithResources
{
    public static void main(String args[])
    {
        try(
            FileOutputStream fileOutputStream = new FileOutputStream
            Stream("abc.txt");
            InputStream input = new FileInputStream("abc.txt")){
            // -----Code to write data into file-----
            String msg = "Welcome to Swati Computers";
            byte byteArray[] = msg.getBytes(); // Converting string into
            byte array
            fileOutputStream.write(byteArray); // Writing data into file

            System.out.println("-----Data written into file-----
            -----");
            System.out.println(msg);

            // -----Code to read data from file-----
            // Creating input stream instance

            DataInputStream inst = new DataInputStream(input);
            int data = input.available();

            //Returns an estimate of the number of bytes that can be
            read from this input stream.
            byte[] byteArray2 = new byte[data]; //
            inst.read(byteArray2);
```

```

String str = new String(byteArray2); // passing byte array
into String constructor
System.out.println("-----Data read from file-----
---");

System.out.println(str); // display file data
}
catch(Exception exception){
System.out.println(exception);
}
}
}
}

```

82. Write a Java class to override method and handle exception?

```

import java.io.*;
class Parent
{
    void msg () throws Exception{System.out.println("parent");}
}

class TestChildException extends Parent
{
    void msg()throws ArithmeticException{System.out.println("child");}

    public static void main(String arg[]){
        Parent p=new TestChildException ();
        try{
            p.msg();
        }catch(Exception e){}
    }
}

```

83. Write a Java program to handle insufficient fund exception when withdrawing amount from bank, create user defined exception?

```
import java.io.*;
```

```
public class InsufficientFundsException extends Exception
{
    private double amount;
```

```
    public InsufficientFundsException(double amount)
    {
        this.amount = amount;
    }
```

```
    public double getAmount()
    {
        return amount;
    }
}
```

```
public class CheckingAccount
{
    private double balance;
    private int number;
```

```
    public CheckingAccount(int number)
    {
        this.number = number;
    }
```

```
    public void deposit(double amount)
    {
        balance += amount;
    }
```

```
    public void withdraw(double amount) throws InsufficientFundsException
```

```
{
    if(amount <= balance)
    {
        balance -= amount;
    }else
    {
        double needs = amount - balance;
        throw new InsufficientFundsException(needs);
    }
}

public double getBalance()
{
    return balance;
}

public int getNumber()
{
    return number;
}
}

public class BankDemo
{

    public static void main(String [] args)
    {
        CheckingAccount c = new CheckingAccount(101);
        System.out.println("Depositing $500...");
        c.deposit(500.00);

        try
        {
            System.out.println("\nWithdrawing $100...");
            c.withdraw(100.00);
            System.out.println("\nWithdrawing $600...");
```

```
        c.withdraw(600.00);
    } catch (InsufficientFundsException e)
    {
        System.out.println("Sorry, but you are short $" + e.getAmount());
        e.printStackTrace();
    }
}
```

PACKAGE

Packages are collection of similar type of classes, interface, and sub packages. It is used to prevent naming conflict. java packages provide access protection. Packages can be built in and user defined.

The **package** keyword is used to create a package in Java.

Compile class in package:

```
javac -d directory javafilename
```

To compile: `javac -d simple.java`

To run: `java mypack.simple`

84. Write a program to create a class in a package and use in another package?

```
package pack;
public class A
{
    public void msg()
    {System.out.println("Hello");}
}
.....
package mypack;
import pack.*;

class B
{
    public static void main(String args[])
    {
        A obj = new A();
        obj.msg();
    }
}
```

85. Write a Java class in package to use variable length argument (avoid overloading) to add many different numbers.?

```
package calci;
class use_packages
{
    int s=0;
    public void sum(int...a) // varArgs
    {
        for(int i : a)
        {
            s+=i;    // s=s+i
        }
        System.out.println("sum="+s);
    }
    public static void main(String [] aw)
    {
        new use_packages().sum(3,4,6);
        new use_packages().sum(3,4,6,6,7,3,2);
        new use_packages().sum(3,4,6,78,34,23,12);
    }
}
```

86. Create a class A in package "a", create another class A1 in subpackage of package "a", create another class B in package "a" and create objects of A and A1 in B class.

```
package a;
public class A
{
    public A()
    {
        System.out.println("A class default constructor");
    }
}

.....

package a.a1;
public class A1
```



```

{
public A1()
{
    System.out.println("A1 class default constructor");
}
}

.....

package a;
import a.*;
import a.a1.*;
class B
{
public static void main(String []dsf)
{
    new A();
    new A1();
    System.out.println("B class");
}
}

```

87. Create a class “First” in package “p” and another class “Third” in package “q”. use First class in Third class?

```

package p;
public class First1
{
    public First1()
    {
        System.out.println("First1 called");
    }
    public String show()
    {
        return " welcome";
    }
}

.....
package p;

```

```
public class Second1
{
    public Second1()
    {
        System.out.println("Second1 called");
    }
    public static void main(String [] arg)
    {
        new First1();
    }
}
```

... ..

```
package q;
import p.*;
import p.p1.*;
class Third1
{
    public static void main(String [] arg)
    {
        First1 f=new First1();
        System.out.println(f.show());
        new Second1();
        new Fourth1();
    }
}
```

88. Create a class “address” and another class” employee”, use reference of “address” in “employee”?

```
public class Address
{
    String city,state,country;

    public Address(String city, String state, String country)
    {
        this.city = city;
        this.state = state;
```

```
        this.country = country;
    }
}
.....
public class Emp
{
    int id;
    String name;
    Address address;

    public Emp(int id, String name,Address address)
    {
        this.id = id;
        this.name = name;
        this.address=address;
    }

    void display()
    {
        System.out.println(id+" "+name);
        System.out.println(address.city+" "+address.state+" "+address.
        country);
    }

    public static void main(String[] args)
    {
        Address address1=new Address("jpr","Raj","india");
        Address address2=new Address("ajmer","Raj","india");

        Emp e=new Emp(111,"Shweta",address1);
        Emp e2=new Emp(112,"Swati",address2);

        e.display();
        e2.display();
    }
}
```

89. Create an interface in package and a class in same package and implement interface?

```
package animals;
```

```
interface Animal
{
    public void eat();
    public void travel();
}
```

```
.....
```

```
package animals;
```

```
public class Mammal implements Animal
{
    public void eat() {
        System.out.println("Mammal eats");
    }
```

```
    public void travel() {
        System.out.println("Mammal travels");
    }
```

```
    public int noOfLegs()
{
    return 0;
}
```

```
public static void main(String args[])
{
    Mammal m = new Mammal();
    m.eat();
    m.travel();
}
```

COLLECTION AND GENERICS

A Collection is a group of individual objects represented as a single unit.

Java provides Collection Framework which defines several classes and interfaces to represent a group of objects as a single unit.

The Collection interface (**java.util.Collection**) and Map interface (**java.util.Map**) are two main root interfaces of Java collection classes.

Collections in Java is a framework that provides an architecture to store and manipulate the group of objects.

All the operations that you perform on a data such as searching, sorting, insertion, manipulation, deletion, and so on can be performed by Java Collections easily.

All collections frameworks contain the following:

Interfaces: These are abstract data types that represent collections. Interfaces allow collections to be manipulated independently of the details of their representation. In object-oriented languages, interfaces generally form a hierarchy.

Implementations, i.e., Classes: These are the concrete implementations of the collection interfaces. In essence, they are reusable data structures.

Algorithms: These are the methods that perform useful computations, such as searching and sorting, on objects that implement collection interfaces. The algorithms are said to be polymorphic: that is, the same method can be used on many different implementations of the appropriate collection interface.

Java Collection framework provides many interfaces (Set, List, Queue, Deque, and so on) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet and so on).

Generics helps us to create a single class, which can be useful to operate on multiple data types. A class, interface or a method that operates on a parameterized type is called generics class, interface or method. Generics adds type safety.

Remember that generics only works on objects, not primitive types.

90. Write a program to add and remove elements in array list?

```
import java.util.*;
public class ArrayListDemo
{
    public static void main(String args[])
    {
        // create an array list
        ArrayList al = new ArrayList();
        System.out.println("Initial size of al: " + al.size());

        // add elements to the array list
        al.add("C");
        al.add("A");
        al.add("E");
        al.add("B");
        al.add("D");
        al.add("F");
        al.add(1, "A2");
        System.out.println("Size of al after additions: " + al.size());

        // display the array list
        System.out.println("Contents of al: " + al);

        // Remove elements from the array list
        al.remove("F");
        al.remove(2);
        System.out.println("Size of al after deletions: " + al.size());
        System.out.println("Contents of al: " + al);
    }
}
```

91. Implement key-value pair in generics?

```
public class Pair<K, V>
{
    private K key;
```

```

private V value;
public Pair(K key, V value)
{
    this.key = key; this.value = value;
}
public void setKey(K key)
{
    this.key = key;
}
public void setValue(V value)
{
    this.value = value;
}
public K getKey()
{
    return key;
}
public V getValue()
{
    return value;
}
}

Pair<Integer, String> p1 = new Pair<>(1, "apple");
Pair<Integer, String> p2 = new Pair<>(2, "pear");

```

92. Create a Hashset and add objects?

```

public class coll
{
    public static void main(String [] ar)
    {
        Integer a1=new Integer(34);
        Set s=new HashSet();
        //Set s=new TreeSet(); //sorted
    }
}

```

```
s.add(new Integer(40));
s.add(new Integer(20));
s.add(new Integer(10));
s.add(new Integer(30));
    System.out.println(s);
}
}
```

93. Create a HashMap with integer key and string value and add elements?

```
public class coll
{
    public static void main(String [] ar)
    {

        Map<Integer,String> m=new HashMap<Integer,String>();
        m.put(1,"A");
        m.put(2,"s");
        m.put(3,"d");
        m.put(4,"c");
        System.out.println(m);
        s=m.keySet();
        System.out.println(s);
        s=m.entrySet();
        System.out.println(s);
        System.out.println(m.get(3));
    }
}
```

94. Use generics to add integer in array list?

```
public class coll
{
    public static void main(String [] ar)
    {
        ArrayList<Integer> a=new ArrayList<Integer>();
```



```

        //ArrayList a=new ArrayList();
        a.add(12);
        a.add(45);
        a.add(89);
        a.add(56);
        a.add(a1);
        // a.add("computer");
        System.out.println(a);
    }
}

```

95. Create a single generic method to print different type of elements of array?

```

public class GenericMethodTest
{
    // generic method printArray
    public static < E > void printArray( E[] inputArray ) {
        // Display array elements
        for(E element : inputArray)
        {
            System.out.printf("%s ", element);
        }
        System.out.println();
    }

    public static void main(String args[])
    {
        // Create arrays of Integer, Double and Character
        Integer[] intArray = { 1, 2, 3, 4, 5 };
        Double[] doubleArray = { 1.1, 2.2, 3.3, 4.4 };
        Character[] charArray = { 'H', 'E', 'L', 'L', 'O' };

        System.out.println("Array integerArray contains:");
        printArray(intArray); // pass an Integer array
    }
}

```

```
        System.out.println("\nArray doubleArray contains:");
        printArray(doubleArray); // pass a Double array

        System.out.println("\nArray characterArray contains:");
        printArray(charArray); // pass a Character array
    }
}
```

96. Create a generic class to add integer and string?

```
public class Box<T>
{
    private T t;
    public void add(T t)
    {
        this.t = t;
    }
    public T get()
    {
        return t;
    }
    public static void main(String[] args)
    {
        Box<Integer> integerBox = new Box<Integer>();
        Box<String> stringBox = new Box<String>();

        integerBox.add(new Integer(10));
        stringBox.add(new String("Hello World"));

        System.out.printf("Integer Value :%d\n\n", integerBox.get());
        System.out.printf("String Value :%s\n", stringBox.get());
    }
}
```

97. Write a Java class to sort student data on the basis of age using comparable interface?

```
class Student implements Comparable<Student>
{
    int rollno;
    String name;
    int age;
    Student(int rollno,String name,int age){
        this.rollno=rollno;
        this.name=name;
        this.age=age;
    }

    public int compareTo(Student st)
    {
        if(age==st.age)
            return 0;
        else if(age>st.age)
            return 1;
        else
            return -1;
    }
}

.....

import java.util.*;
import java.io.*;
public class Sort
{
    public static void main(String args[]){
        ArrayList<Student> al=new ArrayList<Student>();
        al.add(new Student(101,"Vijay",23));
        al.add(new Student(106,"Ajay",27));
        al.add(new Student(105,"Jai",21));
```

```
        Collections.sort(al);
        for(Student st:al){
            System.out.println(st.rollno+" "+st.name+" "+st.age);
        }
    }
}
```

98. Write Java class to add book (class) and print it using priority queue?

```
import java.util.*;
class Book implements Comparable<Book>
{
    int id;
    String name,author,publisher;
    int quantity;
    public Book(int id, String name, String author, String publisher, int
    quantity) {
        this.id = id;
        this.name = name;
        this.author = author;
        this.publisher = publisher;
        this.quantity = quantity;
    }
    public int compareTo(Book b)
    {
        if(id>b.id)
        {
            return 1;
        }
        else if(id<b.id)
        {
            return -1;
        }
        else
        {
```

```

        return 0;
    }
}
}

public class LinkedListExample {
public static void main(String[] args) {
    Queue<Book> queue=new PriorityQueue<Book>();
    //Creating Books
    Book b1=new Book(121,"C question Bank ",swati,"BPB",8);
    Book b2=new Book(233,"java practical solution",swati,"BPB",6);
    //Adding Books to the queue
    queue.add(b1);
    queue.add(b2);
    System.out.println("Traversing the queue elements:");
    //Traversing queue elements
    for(Book b:queue){
        System.out.println(b.id+"    "+b.name+"    "+b.author+"    "+b.
publisher+"    "+b.quantity);
    }
    queue.remove();
    System.out.println("After removing one book record:");
    for(Book b:queue){
        System.out.println(b.id+"    "+b.name+"    "+b.author+"    "+b.
publisher+"    "+b.quantity);
    }
}
}
}

```

99. Write a Java program to display elements of hashtable using enumeration?

```

import java.util.Enumeration;
import java.util.Hashtable;
public class Main {
    public static void main(String[] args) {
        Hashtable ht = new Hashtable();
        ht.put("1", "One");
    }
}

```

```
        ht.put("2", "Two");
        ht.put("3", "Three");
        Enumeration e = ht.elements();

        while(e.hasMoreElements()) {
            System.out.println(e.nextElement());
        }
    }
}
```

100. Write a program to rotate elements of List, depending upon second argument of method?

```
import java.util.*;
public class Rotate_List {
    public static void main(String[] args) {
        List list = Arrays.asList("one Two three Four five six".split(" "));
        System.out.println("List :"+list);
        Collections.rotate(list, 3);
        System.out.println("rotate: " + list);
    }
}
```

101. Write a program to search all the files in directory having file name starting with "s"?

```
import java.io.*;

public class Display_files {
    public static void main(String[] args) {
        File dir = new File("C:");
        FilenameFilter filter = new FilenameFilter() {
            public boolean accept (File dir, String name) {
                return name.startsWith("s");
            }
        };
    }
};
```

```
String[] children = dir.list(filter);
if (children == null) {
    System.out.println("Either dir does not exist or is not a directory");
} else {
    for (int i=0; i< children.length; i++) {
        String filename = children[i];
        System.out.println(filename);
    }
}
}
```

APPLET

Applet is a Java program which can be embed in webpage and runs inside the browser and works at client side.

Lifecycle of Java Applet

1. Applet is initialized. (init())
2. Applet is started. (start())
3. Applet is painted. (paint())
4. Applet is stopped. (stop())
5. Applet is destroyed. (destroy())

AWT package:

AWT stands for Abstract window toolkit.

Graphics: color, font etc.

UI components: Button, TextField, List etc.

Layouts: FlowLayout, BorderLayout (for arranging objects/controls)

Container: Frame, Dialog, Panel

Events: Listener – ActionListener, MouseListener, WindowListener etc.

Listener: Listener listen the event which associated with the application.

This can be context or session of the application. Listener gives us a great control over application without touching architecture of the application.

102. Write an applet to display welcome message and create HTML to run it?

```
import java.applet.Applet;
import java.awt.Graphics;
public class First extends Applet
{

    public void paint(Graphics g)
```



```
{
g.drawString("welcome to swati computers",150,150);
}
}

.. . . .
<html>
<body>
<applet code="First.class" width="300" height="300">
</applet>
</body>
</html>
```

103. Write a Java applet to set background color and input name as parameter?

```
import java.applet.Applet;
import java.awt.*;
/*<applet code="app1.class" height="200" width="300" hspace=10
vspace=23>
<param name="surName" value="saxena">
</applet>
*/
public class app1 extends Applet
{
Font f;
String s;
Color c;
public void init()
{
c=new Color(456812);
s=getParameter("surName");
f=new Font("Times New Roman",67,24);
setBackground(Color.red);
}
public void paint(Graphics g)
{
```

```
        g.setColor(c);
        g.setFont(f);
        g.drawString("hello subroto",50,50);
        g.drawString(s,200,50);
    }
}
```

104. Write an applet to draw a red line, one green rectangle, one pink, and one blue square?

```
import java.applet.Applet;
import java.awt.*;
/*<applet code="app1.class" height="200" width="300" hspace=10
vspace=23>
  <param name="surName" value="saxena">
</applet>
*/
public class app1 extends Applet
{
    Font f;
    String s;
    Color c;
    public void init()
    {
        c=new Color(456812);
        s=getParameter("surName");
        f=new Font("Times New Roman",67,24);
        setBackground(Color.red);
    }
    public void paint(Graphics g)
    {
        g.setColor(c);
        g.setFont(f);
        g.drawString("hello subroto",50,50);
        g.drawString(s,200,50);
    }
}
```

105. Write an applet for login box, with user name , password , and login button?

```
import java.applet.Applet;
import java.awt.*;
/*<applet code="app3.class" height=500 width=500>
</applet>
*/
public class app3 extends Applet
{
    Label lname,lpwd;
    TextField tn,tp;
    Button signin;
    public void init()
    {
        lname=new Label("Enter ur name");
        lpwd=new Label("Enter password");
        tn=new TextField(10);
        tp=new TextField(10);
        tp.setEchoChar('*');
        signin=new Button("SignIn");
        add(lname);
        add(tn);
        add(lpwd);
        add(tp);
        add(signin);
    }
}
```

106. Write an applet to input user name, password, gender, course, select country from list and register form?

```
import java.applet.Applet;
import java.awt.*;
/*<applet code="myap2.class" height=300 width="300"></applet>*/
public class myap2 extends Applet
```

```
{
    Label un, lbl_pwd;
    TextField tun, txt_pwd;
    Button signIn, signUp;
    Checkbox c, cpp, java, male, female;
    Choice list;
    //FlowLayout flo;
    GridLayout glo;
    public void init()
    {
        //flo=new FlowLayout(FlowLayout.CENTER);
        glo=new GridLayout(6,2,10,10);
        setLayout(glo);
        un=new Label("Enter User Name");
        lbl_pwd=new Label("Enter password");
        tun=new TextField(10);
        txt_pwd=new TextField(10);
        txt_pwd.setEchoChar('*');
        signIn=new Button("SignIn");
        signUp=new Button("Register");
        CheckboxGroup cbg=new CheckboxGroup();
        male=new Checkbox("male",cbg,true);
        female=new Checkbox("female",cbg,false);
        list=new Choice();
        list.addItem("India");
        list.addItem("NZ");
        list.addItem("SA");
        list.addItem("Eng");
        add(un);
        add(tun);
        add(lbl_pwd);
        add(txt_pwd);
        add(c=new Checkbox("C"));
        add(cpp=new Checkbox("C++"));
        add(java=new Checkbox("JAVA"));
```

```

        add(male);
        add(female);
        add(list);
        add(signIn);
        add(signUp);
    }
}

```

107. Create applet, draw a line and four color circle at both end of lines?

```

import java.applet.Applet;
import java.awt.*;
/*
<applet code="myapp1.class" height="500" width="550">
<param name="myname" value="surabhi">
</applet>
*/
public class myapp1 extends Applet
{
    public void paint(Graphics g)
    {
        g.setColor(Color.blue);
        g.fillArc(100,100,100,100,0,90);
        g.setColor(Color.green);
        g.drawArc(100,100,100,100,90,90);
        g.fillArc(100,100,100,100,90,90);
        g.setColor(Color.yellow);
        g.fillArc(100,100,100,100,180,90);
        g.setColor(Color.red);
        g.fillArc(100,100,100,100,270,90);

        g.setColor(Color.blue);
        g.fillArc(400,100,100,100,0,90);
        g.setColor(Color.green);
        g.drawArc(400,100,100,100,90,90);
    }
}

```

```
g.fillArc(400,100,100,100,90,90);
g.setColor(Color.yellow);
g.fillArc(400,100,100,100,180,90);
g.setColor(Color.red);
g.fillArc(400,100,100,100,270,90);
g.setColor(Color.black);
g.drawLine(150,200,300,350);
g.drawLine(450,200,300,350);
}
}
```

108. Draw a four coloured ball and a polygon in applet?

```
import java.applet.Applet;
import java.awt.*;
/*<applet code="poly.class" height=400 width=400></applet>*/
public class poly extends Applet
{
    public void paint(Graphics g)
    {
        g.setColor(Color.red);
        int []x=new int[]{50,100,150} ;
        int []y=new int[]{150,50,150};
        g.fillPolygon(x,y,3);
        g.setColor(Color.green);
        g.fillArc(50,300,100,100,0,90);
        g.setColor(Color.blue);
        g.fillArc(50,300,100,100,90,90);
        g.setColor(Color.yellow);
        g.fillArc(50,300,100,100,180,90);
        g.setColor(Color.pink);
        g.fillArc(50,300,100,100,270,90);
    }
}
```

109. Draw india (national) flag in applet?

```
import java.applet.*;
import java.awt.*;
/*<applet code="Flag.class" height=300 width=300 ></applet>*/
public class Flag extends Applet
{
    public void paint(Graphics g)
    {
        g.fillOval(60,450,120,50);
        g.fillRect(110,60,10,400);
        g.setColor(Color.red);
        g.fillRect(120,80,150,30);
        g.setColor(Color.white);
        g.fillRect(120,110,150,30);
        g.setColor(Color.green);
        g.fillRect(120,140,150,30);
        g.setColor(Color.black);
        g.drawRect(120,80,150,90);
        g.setColor(Color.blue);

        g.drawOval(180,110,30,30);
        int t=0;
        int x=195,y=125;
        double x1,y1;
        double r=15;
        double d;
        for(int i=1;i<=24;i++)
        {
            d=(double)t*3.14/180.0;
            x1=x+(double)r*Math.cos(d);
            y1=y+(double)r*Math.sin(d);
            g.drawLine(x,y,(int)x1,(int)y1);
            t+=360/24;
        }
    }
}
```

110. Draw a color box of nine colors in applet?

```
import java.applet.*;
import java.awt.*;
/*<applet code="Colors.class" height=300 width=300 ></applet>*/
public class Colors extends Applet
{
    public void paint(Graphics g)
    {
        g.setColor(Color.red);
        g.fillRect(0,0,40,40);
        g.setColor(Color.pink);
        g.fillRect(50,0,40,40);
        g.setColor(Color.green);
        g.fillRect(100,0,40,40);

        g.setColor(Color.blue);
        g.fillRect(0,50,40,40);
        g.setColor(Color.white);
        g.fillRect(50,50,40,40);
        g.setColor(Color.yellow);
        g.fillRect(100,50,40,40);

        g.setColor(Color.black);
        g.fillRect(0,100,40,40);
        g.setColor(Color.green);
        g.fillRect(50,100,40,40);
        g.setColor(Color.gray);
        g.fillRect(100,100,40,40);
    }
}
```

111. Create a student panel to add total marks and average?

```
import java.awt.*;
import java.applet.*;
```



```
import java.awt.event.*;
/* <applet code="Flag.class" width=400 height=400>
</applet> */
```

public class Flag extends Applet implements

ActionListener,ItemListener

```
{
    String s1,s2,s3;
    TextField t3,t4,t5,t6,t7;
    Button tot,avg;
    Checkbox c1,c2,c3,c4,m,f;
    CheckboxGroup cbg;
    Panel p1,p2,p3,p4;

    public void init()
    {
        s3="";
        tot=new Button("Total");
        avg=new Button("Average");
        c1=new Checkbox("MCA",true);
        c2=new Checkbox("Msc Comp");
        c3=new Checkbox("MSIT");
        c4=new Checkbox("MSIS");

        cbg=new CheckboxGroup();

        m=new Checkbox("Male",cbg,false);
        f=new Checkbox("Female",cbg,true);
        p1=new Panel();
        p1.setLayout(new GridLayout(2,2));

        p1.add(new Label("Student Number "));
        p1.add(new TextField(5));
```

```
p1.add(new Label("Student Name "));
p1.add(new TextField(15));

add(p1);
p2=new Panel(); p2.setLayout(new GridLayout(1,3));
p2.add(new Label("Gender"));
p2.add(m);
p2.add(f);
add(p2);

p3=new Panel(); p3.setLayout(new GridLayout(1,5));
p3.add(new Label("Degree"));
p3.add(c1); p3.add(c2); p3.add(c3); p3.add(c4);

add(p3);

p4=new Panel(); p4.setLayout(new GridLayout(6,2));
p4.add(new Label("Marks in JAVA"));
t3=new TextField(3); p4.add(t3);

p4.add(new Label("Marks in VB .Net"));
t4=new TextField(3); p4.add(t4);

p4.add(new Label("Marks In C"));
t5=new TextField(3); p4.add(t5);

p4.add(new Label("Total "));
t6=new TextField(3); p4.add(t6);

p4.add(new Label(" Average"));
t7=new TextField(3); p4.add(t7);

p4.add(tot); p4.add(avg);

tot.addActionListener(this);
```

```
avg.addActionListener(this);

c1.addItemListener(this);
c2.addItemListener(this);

c3.addItemListener(this);
c4.addItemListener(this);
m.addItemListener(this);
f.addItemListener(this);

add(p4);
}

public void paint(Graphics g)
{
    int no,m1,m2,m3,tot;
    float avg=0.0f;
    no=m1=m2=m3=tot=0;
    try
    {
        m1=Integer.parseInt(t3.getText());
        m2=Integer.parseInt(t4.getText());
        m3=Integer.parseInt(t5.getText());
    }
    catch(Exception e)
    {
    }
    tot=m1+m2+m3;
    avg= tot/3;
    s1=String.valueOf(tot);
    s2=String.valueOf(avg);
}

public boolean action(Event e,Object o)
{
```

```
    repaint();
    return true;
}
public void actionPerformed(ActionEvent e)
{
    s3=e.getActionCommand();
    if(s3.equals("Total"))
        t6.setText(s1);
    if(s3.equals("Average"))
        t7.setText(s2);
    repaint();
}
public void itemStateChanged(ItemEvent e)
{
    repaint();
}
}
```

112. Write an applet to display image ?

```
import java.applet.Applet;
import java.awt.*;
/*<applet code="imgs.class" height=500 width=500>
</applet>
*/
public class imgs extends Applet
{
    Image img;
    public void init()
    {
        img=getImage(getDocumentBase(),"1.jpg");
    }
    public void paint(Graphics g)
    {
        g.drawImage(img,60,90,this);
    }
}
```

113. Create a Frame with four buttons, on button click change the elements of center?

```
import java.awt.*;
import java.awt.event.*;
class fra4 extends Frame implements ActionListener
{
    Button b1,b2,b3,b4;
    Panel p,p1,p2,p3,p4;
    CardLayout cl;
    CheckboxGroup cbg;
    Checkbox m,f;
    Choice list;
    public fra4(String tit)
    {
        super(tit);
        add(b1=new Button("1"),"North");
        add(b2=new Button("2"),"West");
        add(b3=new Button("3"),"East");
        add(b4=new Button("4"),"South");
        p=new Panel();
        p1=new Panel();
        p2=new Panel();
        p3=new Panel();
        p4=new Panel();
        cl=new CardLayout();
        list=new Choice();
        list.addItem("Red");
        list.addItem("blue");
        list.addItem("green");
        list.addItem("white");
        p.setLayout(cl);
        cbg=new CheckboxGroup();
        m=new Checkbox("male",cbg,true);
        f=new Checkbox("female",cbg,false);
        p1.add(list);
```

```
p2.add(new Checkbox("hi"));
p3.add(new Button("Click"));
p4.add(m);
p4.add(f);
p.add(p1,"1");
p.add(p2,"2");
p.add(p3,"3");
p.add(p4,"4");
add(p);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);

setVisible(true);
setSize(400,400);
}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==b1)
        cl.show(p,"1");
    else if(ae.getSource()==b2)
        cl.show(p,"2");
    else if(ae.getSource()==b3)
        cl.show(p,"3");
    else if(ae.getSource()==b4)
        cl.show(p,"4");
}
}
class fra4_demo
{
    public static void main(String [] arg)
    {
        new fra4("hi");
    }
}
```

114. Draw a frame and on click of radio button give message?

```
import java.awt.event.*;
import java.awt.*;
class fra7 extends Frame implements ItemListener
{
    Checkbox m,f;
    CheckboxGroup cbg;
    Label res;
    public fra7(String tit)
    {
        super(tit);
        setLayout(new GridLayout(2,2,10,10));
        cbg=new CheckboxGroup();
        m=new Checkbox("male",cbg,true);
        f=new Checkbox("female",cbg,false);
        res=new Label("Message");
        add(m);
        add(f);
        add(res);
        m.addItemListener(this);
        f.addItemListener(this);
        setSize(200,200);
        setVisible(true);
    }
    public void itemStateChanged(ItemEvent ie)
    {
        if(m.getState()==true)
            res.setText("Hello man");
        else if(f.getState()==true)
            res.setText("Hello maam");
    }
}
class fra7_demo
{
```

```
public static void main(String [] rag)
{
    new fra7("hi");
}
}
```

115. Draw a frame to implement KeyListener, on type or press the code of key should display?

```
import java.awt.event.*;
import java.awt.*;
class fra8 extends Frame implements KeyListener
{
    int x;
    char c;
    Label res;

    public fra8(String tit)
    {
        super(tit);
        setLayout(new FlowLayout());
        addKeyListener(this);
        add(res=new Label("Answer"));
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent we)
            {
                System.exit(0);
            }
            public void windowIconified(WindowEvent we)
            {
                setVisible(false);
            }
        });
        setSize(300,300);
        setVisible(true);
    }
}
```



```

    }
    public void keyPressed(KeyEvent ke)
    {
        x=ke.getKeyCode();
        res.setText("");
        res.setText(""+x);
    }
    public void keyReleased(KeyEvent ke)
    {}
    public void keyTyped(KeyEvent ke)
    {
        c=ke.getKeyChar();
        res.setText("");
        res.setText(""+c);
    }
}
}
class fra8_demo
{
    public static void main(String [] arf)
    {
        new fra8("hi");
    }
}

```

116. Draw a frame to implement itemLisener on checkbox and radio button?

```

import java.awt.*;
import java.awt.event.*;
class scomp1 extends Frame implements ItemListener
{
    //Button scom1;
    Label basic,java,develop,combo,scom,chos,chos1,chos2;
    Checkbox core,adv,comp,c,cpp,comb1,comb2;
    CheckboxGroup b1,cp1;
    Choice l;
}

```

```
public scomp1(String tt)
{
    super(tt);
    setLayout(null);
    scom=new Label("SWATI COMPUTERS");
    chos=new Label("");
    chos1=new Label("");
    chos2=new Label("");
    l=new Choice();
    cp1=new CheckboxGroup();
    b1=new CheckboxGroup();
    basic=new Label("basic");
    java=new Label("java");
    develop=new Label("developing");
    combo=new Label("combo packs");
    core=new Checkbox("Core");
    adv=new Checkbox("Advance");
    comp=new Checkbox("Complete");
    c=new Checkbox("C",b1,true);
    cpp=new Checkbox("C++",b1,false);
    comb1=new Checkbox("C,C++,Core",cp1,true);
    comb2=new Checkbox("Core,Advance",cp1,false);
    l.add("java");
    l.add(".net");
    l.add("pta nii");
    chos.setBounds(150,500,300,20);
    chos1.setBounds(150,520,300,20);
    chos2.setBounds(150,540,300,20);
    scom.setBounds(250,100,150,30);
    basic.setBounds(100,150,150,20);
    java.setBounds(400,150,50,20);
    develop.setBounds(100,300,150,20);
    combo.setBounds(400,300,150,20);
    core.setBounds(400,230,100,20);
    adv.setBounds(400,250,100,20);
```

```

        comp.setBounds(400,210,100,20);
        c.setBounds(100,210,150,20);
        cpp.setBounds(100,230,150,20);
        comb1.setBounds(400,320,150,20);
        comb2.setBounds(400,340,150,20);
        l.setBounds(100,320,150,20);
        add(scom);
        add(basic);
        add(java);
        add(core);
        add(adv);
        add(comp);
        add(develop);
        add(combo);
        add(c);
        add(l);
        add(cpp);
        add(comb1);
        add(comb2);
        add(chos);
        add(chos1);
        add(chos2);
        c.addItemListener(this);
        cpp.addItemListener(this);
        core.addItemListener(this);
        adv.addItemListener(this);
        comb1.addItemListener(this);
        comb2.addItemListener(this);
        setVisible(true);
        setSize(200,200);
    }
    public void itemStateChanged(ItemEvent ie)
    {
        if(c.getState()==true)
            chos.setText("you have chosen C language in basics ");
    }

```

```
if(cpp.getState()==true)
chos.setText("you have chosen C++ language in basics");
if(comb1.getState()==true)
chos1.setText("& you have chosen a combo of C,c++,core languages ");
if(comb2.getState()==true)
chos1.setText("& you have chosen a combo of core ,advance languages");
```

```
if(comp.getState()==true)
chos2.setText("you have chosen Complete part in java ");
if(core.getState()==true)
chos2.setText("you have chosen Core part in java ");
if(adv.getState()==true)
chos2.setText("& you have chosen advance part in java ");
```

```
}
}
class scomp
{
public static void main(String [] rag)
{
    new scomp1("hi");
}
}
```

117. Draw a frame to implement `MouseListener` and `MouseMotionListener` to display the position of mouse?

```
import java.awt.*;
import java.awt.event.*;
class myfr7 extends Frame implements ActionListener,MouseListener,
MouseMotionListener
{
    Button b1,b2,b3,b4;
    TextField t;
    Label m;
    public myfr7(String tit)
    {
        super(tit);
```

```
setLayout(null);
m=new Label("");
t=new TextField(20);
b1=new Button("1");
b2=new Button("2");
b3=new Button("3");
b4=new Button("4");
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
t.setBounds(50,200,150,30);
b1.setBounds(30,50,50,20);
b2.setBounds(100,50,50,20);
b3.setBounds(30,100,50,20);
b4.setBounds(100,100,50,20);
m.setBounds(200,200,100,30);
addMouseListener(this);
addMouseMotionListener(this);
add(t);
add(b1);
add(b2);
add(b3);
add(b4);
add(m);

addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent we)
    {
        dispose();
    }
});

setSize(300,300);
setVisible(true);
```

```
}
public void mouseMoved(MouseEvent me)
{
    t.setText("position: x="+me.getX()+",y="+me.getY());
}
public void mouseDragged(MouseEvent me){}
public void mouseClicked(MouseEvent me)
{
    t.setText("Clicked");
}
public void mouseEntered(MouseEvent me)
{
    t.setText("Entered");
}
public void mouseExited(MouseEvent me)
{
    t.setText("Exited");
}
public void mousePressed(MouseEvent me)
{
    t.setText("pressed");
}
public void mouseReleased(MouseEvent me)
{
    m.setText("Released");
}

public void actionPerformed(ActionEvent ae)
{
}
}
class frm7
{
    public static void main(String [] arg)
    {
```

```

        new myfr7("hi");
    }
}

```

118. Draw a frame to implement itemListener and display selected course from checkbox?

```

import java.awt.*;
import java.awt.event.*;
class myfr8 extends Frame implements ItemListener
{
    Checkbox c,cpp;
    int i;
    Label msg;
    String s="",s1="";
    public myfr8(String tit)
    {
        super(tit);
        setLayout(new FlowLayout());
        add(c=new Checkbox("C"));
        add(cpp=new Checkbox("C++"));
        add(msg=new Label("Nothing Selected"));
        c.addItemListener(this);
        cpp.addItemListener(this);
        setSize(200,200);
        setVisible(true);
    }
    public void itemStateChanged(ItemEvent ie)
    {
        if(ie.getSource()==c )
            if( c.getState()==true)
                s=s+"C";
            else
                s="";
        if(ie.getSource()==cpp)
            if( cpp.getState()==true)

```

```
        s1=s1+"cpp";
    else
        s1="";
    msg.setText(s +" "+ s1);
}
}
class frm8
{

    static public void main(String [] rag)
    {
        new myfr8("hi");
    }
}
```

119. Write an applet to move image up, down, left, right on button click?

```
import java.applet.Applet;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code="imgs12.class" height=500 width=500>
</applet>
*/
public class imgs12 extends Applet implements

    ActionListener,KeyListener
    {
        Image img;
        int x,y=110,z=110,tn=110;
        Label res;
        Scanner sc;
        Button left,right,up,down;
        public void init()
        {
            img=getImage(getDocumentBase(),"12.jpg");
```



```

    left=new Button("Left");
    right=new Button("right");
    up=new Button("up");
    down=new Button("down");
    add(res=new Label(""));
    add(left);
    add(right);
    add(up);
    add(down);
    left.addActionListener(this);
    up.addActionListener(this);
    down.addActionListener(this);
    right.addActionListener(this);
    addKeyListener(this);
}
public void keyTyped(KeyEvent ke){}
public void keyPressed(KeyEvent ke)
{
    y=ke.getKeyCode();
    //res.setText(""+y);
    if(y==37)
        z=z-10;
    if(y==38)
        tn=tn-10;
    if(y==39)
        z=z+10;
    if(y==40)
        tn=tn+10;
    repaint();
}
public void keyReleased(KeyEvent ke){}

public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==left)

```

```
        z=z-10;
        if(ae.getSource()==up)
            tn=tn-10;
        if(ae.getSource()==right)
            z=z+10;
        if(ae.getSource()==down)
            tn=tn+10;
        repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString(""+y,100,100);
        g.drawImage(img,z,tn,this);
    }
}
```

120. Write a program to create MenuBar with MenuItem and sub menu?

```
import java.awt.*;
class MenuExample
{
    MenuExample(){
        Frame f= new Frame("Menu and MenuItem Example");
        MenuBar mb=new MenuBar();
        Menu menu=new Menu("Menu");
        Menu submenu=new Menu("Sub Menu");
        MenuItem i1=new MenuItem("Item 1");
        MenuItem i2=new MenuItem("Item 2");
        MenuItem i3=new MenuItem("Item 3");
        MenuItem i4=new MenuItem("Item 4");
        MenuItem i5=new MenuItem("Item 5");
        menu.add(i1);
        menu.add(i2);
        menu.add(i3);
        submenu.add(i4);
        submenu.add(i5);
    }
}
```

```

        menu.add(submenu);
        mb.add(menu);
        f.setMenuBar(mb);
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main(String args[])
    {
        new MenuExample();
    }
}

```

121. Write a program to create popup menu?

```

import java.awt.*;
import java.awt.event.*;
class PopupMenuExample
{
    PopupMenuExample(){
        final Frame f= new Frame("PopupMenu Example");
        final PopupMenu popupmenu = new PopupMenu("Edit");
        MenuItem cut = new MenuItem("Cut");
        cut.setActionCommand("Cut");
        MenuItem copy = new MenuItem("Copy");
        copy.setActionCommand("Copy");
        MenuItem paste = new MenuItem("Paste");
        paste.setActionCommand("Paste");
        popupmenu.add(cut);
        popupmenu.add(copy);
        popupmenu.add(paste);
        f.addMouseListener(new MouseAdapter()
        {
            public void mouseClicked(MouseEvent e)
            {
                popupmenu.show(f , e.getX(), e.getY());
            }
        });
    }
}

```

```
        }
    });
    f.add(popupmenu);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
}
public static void main(String args[])
{
    new PopupMenuExample();
}
}
```

Adapter Class

Adapter classes provide the default implementation of listener interfaces.

If you inherit the adapter class, you will not be forced to provide the implementation of all the methods of listener interfaces. So it saves code.

122. Write a program to draw circle on mouse click?

```
import java.awt.*;
import java.awt.event.*;
public class C_Mouse extends MouseAdapter
{
    Frame f;
    C_Mouse(){
        f=new Frame("Mouse Adapter");
        f.addMouseListener(this);

        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
    public void mouseClicked(MouseEvent e)
    {
```

```

        Graphics g=f.getGraphics();
        g.setColor(Color.BLUE);
        g.fillOval(e.getX(),e.getY(),30,30);
    }
    public static void main(String[] args) {
        new C_Mouse();
    }
}

```

123. Write a program to open dialog box on click button?

```

import java.awt.*;
import java.awt.event.*;

public class Ex_Dialog extends WindowAdapter implements

    ActionListener
    {
        Frame frame;
        Label label1;
        TextField field1;
        Button button1, button2, button3;
        Dialog d1, d2, d3;

        Ex_Dialog()
        {
            frame = new Frame("Frame");
            button1 = new Button("Open Modal Dialog");

            label1 = new Label("Click on the button to open a Modal Dialog");

            frame.add(label1);
            frame.add(button1);

            button1.addActionListener(this);
            frame.pack();

```

```
frame.setLayout(new FlowLayout());
frame.setSize(330,250);
frame.setVisible(true);
}

public void actionPerformed(ActionEvent ae)
{

if(ae.getActionCommand().equals("Open Modal Dialog"))
{
//Creating a non-modeless blocking Dialog
d1= new Dialog(frame,"Modal Dialog",true);
Label label= new Label("You must close this dialog window to use

Frame window",Label.CENTER);
d1.add(label);

d1.addWindowListener(this);
d1.pack();
d1.setLocationRelativeTo(frame);
d1.setLocation(new Point(100,100));
d1.setSize(400,200);
d1.setVisible(true);
}
}

public void windowClosing(WindowEvent we)
{
d1.setVisible(false);
}

public static void main(String...ar)
{
new Ex_Dialog();
}
}
```

124. Write a program to use cardLayout and change buttons on each click?

```
import java.awt.*;
import java.awt.event.*;

import javax.swing.*;

public class CardLayoutExample extends JFrame implements

ActionListener{
    CardLayout card;
    JButton b1,b2,b3;
    Container c;
    CardLayoutExample(){

        c=getContentPane();
        card=new CardLayout(40,30);

        c.setLayout(card);

        b1=new JButton("C");
        b2=new JButton("C++");
        b3=new JButton("JAVA");
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);

        c.add("a",b1);c.add("b",b2);c.add("c",b3);

        public void actionPerformed(ActionEvent e)
        {
            card.next(c);
        }

        public static void main(String[] args)
```

```
{
    CardLayoutExample cl=new CardLayoutExample();
    cl.setSize(400,400);
    cl.setVisible(true);
    cl.setDefaultCloseOperation(EXIT_ON_CLOSE);
}
```

125. Write a Java program to change the card (cardlayout) in middle of border layout on click buttons?

Students please create it yourself..

126. Create a small calculator in Java frame?

```
import java.awt.event.*;
import java.awt.*;
import java.awt.event.WindowEvent;

class myfra1 extends Frame implements ActionListener
{
    TextField t;
    Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0,bplus,bequal,bminus,bmulti,bdiv,bclear;
    char op;
    String val,val1;
    int ans;
    Panel p;
    public myfra1(String tit)
    {
        super(tit);
        setLayout(new FlowLayout());
        t=new TextField(10);
        b1=new Button("1");
        b2=new Button("2");
        b3=new Button("3");
        b4=new Button("4");
```



```
b5=new Button("5");
b6=new Button("6");
b7=new Button("7");
b8=new Button("8");
b9=new Button("9");
b0=new Button("0");
bclear=new Button("C");
p=new Panel();
p.setLayout(new GridLayout(4,4));
bplus=new Button("+");
bminus=new Button("-");
bequal=new Button("=");
bmulti=new Button("*");
bdiv=new Button("/");
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
b6.addActionListener(this);
b7.addActionListener(this);
b8.addActionListener(this);
b9.addActionListener(this);
b0.addActionListener(this);
bclear.addActionListener(this);
bplus.addActionListener(this);
bminus.addActionListener(this);
bmulti.addActionListener(this);
bdiv.addActionListener(this);
bequal.addActionListener(this);
add(t);

p.add(b1);p.add(b2);p.add(b3);p.add(b4);p.add(b5);p.add(b6);p.add(b7);
p.add(b8);p.add(b9);p.add(b0);
```

```
p.add(bplus);p.add(bminus);p.add(bmulti);p.add(bdiv);p.
add(bequal);p.add(bcLEAR);
add(p);
addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});
setLocation(300,300);
setSize(100,200); //frame class function to set height,width of frame
setVisible(true);
}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==b1)
    {
        t.setText(t.getText()+"1");
    }
    if(ae.getSource()==b2)
    {
        t.setText(t.getText()+"2");
    }
    if(ae.getSource()==b3)
    {
        t.setText(t.getText()+"3");
    }
    if(ae.getSource()==b4)
    {
        t.setText(t.getText()+"4");
    }
    if(ae.getSource()==b5)
    {
        t.setText(t.getText()+"5");
    }
    if(ae.getSource()==b6)
```

```
{
t.setText(t.getText()+"6");
}
if(ae.getSource()==b7)
{
t.setText(t.getText()+"7");
}
if(ae.getSource()==b8)
{
t.setText(t.getText()+"8");
}
if(ae.getSource()==b9)
{
t.setText(t.getText()+"9");
}
if(ae.getSource()==b0)
{
t.setText(t.getText()+"0");
}
if(ae.getSource()==bplus)
{
op='+';
val=t.getText();
t.setText("");
}
if(ae.getSource()==bminus)
{
op='-';
val=t.getText();
t.setText("");
}
if(ae.getSource()==bmulti)
{
op='*';
val=t.getText();
```

```
t.setText("");
}
if(ae.getSource()==bclear)
{
t.setText("");
op='';
}
if(ae.getSource()==bdiv)
{
op='/';
val=t.getText();
t.setText("");
}
if(ae.getSource()==bequal)
{
val1=t.getText();
switch(op)
{
    case '+':
        ans=Integer.parseInt(val)+Integer.parseInt(val1);
        break;
    case '-':
        ans=Integer.parseInt(val)-Integer.parseInt(val1);
        break;
    case '*':
        ans=Integer.parseInt(val)*Integer.parseInt(val1);
        break;
    case '/':
        ans=(int)(Integer.parseInt(val)/Integer.parseInt(val1));
        break;
}
t.setText(String.valueOf(ans));
}
}
}
```

```
public class calci1 {
    public static void main(String [] ar)
    {
        new myfra1("calculator");
    }
}
```

127. Write an applet, use GridBagLayout and show the state of checkbox on select?

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
<applet code="GridBagDemo.class" width=250 height=200>
</applet>
*/
public class GridBagDemo extends Applet
implements ItemListener {
    String msg = "";
    Checkbox winXP, winVista, solaris, mac;
    public void init() {
        GridBagLayout gbag = new GridBagLayout();
        GridBagConstraints gbc = new GridBagConstraints();
        setLayout(gbag);
        // Define check boxes.
        winXP = new Checkbox("Windows XP ", null, true);
        winVista = new Checkbox("Windows Vista");
        solaris = new Checkbox("Solaris");
        mac = new Checkbox("Mac OS");
        // Define the grid bag.
        // Use default row weight of 0 for first row.
        gbc.weightx = 1.0; // use a column weight of 1
        gbc.ipadx = 200; // pad by 200 units
        gbc.insets = new Insets(4, 4, 0, 0); // inset slightly from top left
```

```
gbc.anchor = GridBagConstraints.SOUTHWEST;
gbc.gridwidth = GridBagConstraints.NONE;
gbag.setConstraints(winXP, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
gbag.setConstraints(winVista, gbc);
// Give second row a weight of 1.
gbc.weighty = 1.0;
```

```
gbc.gridwidth = GridBagConstraints.RELATIVE;
gbag.setConstraints(solaris, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
gbag.setConstraints(mac, gbc);
// Add the components.
add(winXP);
add(winVista);
add(solaris);
add(mac);
// Register to receive item events.
winXP.addItemListener(this);
winVista.addItemListener(this);
solaris.addItemListener(this);
mac.addItemListener(this);
}
// Repaint when status of a check box changes.
public void itemStateChanged(ItemEvent ie) {
    repaint();
}
// Display current state of the check boxes.
public void paint(Graphics g) {
    msg = "Current state: ";
    g.drawString(msg, 6, 80);
    msg = " Windows XP: " + winXP.getState();
    g.drawString(msg, 6, 100);
    msg = " Windows Vista: " + winVista.getState();
    g.drawString(msg, 6, 120);
```

```

msg = " Solaris: " + solaris.getState();
g.drawString(msg, 6, 140);
msg = " Mac: " + mac.getState();
g.drawString(msg, 6, 160);
}
}

```

128. Create a Label on Frame and move it on pressing arrow keys on keyboard?

```

import java.awt.*;
import java.awt.event.*;
class myfr4 extends Frame implements KeyListener
{
    char ch;
    Label charc,code,move;
    int x,y;
    public myfr4(String tit)
    {
        super(tit);
        x=100;y=100;
        //setLayout(new FlowLayout());
        setLayout(null);
        charc=new Label("Character pressed");
        code=new Label("code");
        move=new Label("moving");
        charc.setBounds(10,10,50,20);
        code.setBounds(70,10,50,20);
        move.setBounds(x,y,40,20);
        add(move);
        addKeyListener(this);
        setSize(300,300);
        setVisible(true);
    }
    public void keyTyped(KeyEvent ke)
    {

```

```
        ch=ke.getKeyChar();
        charc.setText("Character:" +ch);
    }
    public void keyPressed(KeyEvent ke)
    {
        int w=ke.getKeyCode();
        if(w==KeyEvent.VK_UP){code.setText("up          key");move.
        setBounds(x,y-=10,50,20);}
        else          if(w==KeyEvent.VK_DOWN){code.setText("down
        key");move.setBounds(x,y+=10,50,20);}
        else  if(w==KeyEvent.VK_LEFT){code.setText("left  key");move.
        setBounds(x-=10,y,50,20);}
        else if(w==KeyEvent.VK_RIGHT){code.setText("right key");move.
        setBounds(x+=10,y,50,20);}
        else code.setText("Code:" +w);
    }
    public void keyReleased(KeyEvent ke)
    {}
    }
    class myfr4_demo
    {
    public static void main(String [] ar)
    {
        new myfr4("hi");
    }
    }
```

129. Draw a Java Frame to display the position of mouse use MouseListener?

```
import java.awt.*;
import java.awt.event.*;
class myfr3 extends Frame implements MouseListener
{
    Label p,status,r;
    int x,y;
    public myfr3(String tit)
```



```
{
    super(tit);
    setLayout(new FlowLayout());
    add(status=new Label("Status"));
    add(p=new Label("position"));
    add(r=new Label("hello"));
    addMouseListener(this);
    setSize(300,300);
    setVisible(true);
}

public void mouseClicked(MouseEvent me)
{
    status.setText("clicked");
    x=me.getX();
    y=me.getY();
    p.setText("x:"+x+",y:"+y);
}

public void mouseEntered(MouseEvent me)
{
    status.setText("entered");
    x=me.getX();
    y=me.getY();
    p.setText("x:"+x+",y:"+y);
}

public void mouseExited(MouseEvent me)
{
    status.setText("exited");
    x=me.getX();
    y=me.getY();
    p.setText("x:"+x+",y:"+y);
}

public void mousePressed(MouseEvent me)
{
    status.setText("pressed");
    x=me.getX();
```

```
        y=me.getY();
        p.setText("x:"+x+",y:"+y);
    }
    public void mouseReleased(MouseEvent me)
    {
        r.setText("released");
        x=me.getX();
        y=me.getY();
        p.setText("x:"+x+",y:"+y);
    }
}
class myfr3_demo
{
    public static void main(String [] ra)
    {
        new myfr3("hi");
    }
}
```

130. Draw a frame to change the text of textbox when writing into another textbox and close Frame using windowListener?

```
import java.awt.*;
import java.awt.event.*;
class myfr2 extends Frame implements TextListener ,WindowListener
{
    TextField t1,t2;
    public myfr2(String tit)
    {
        super(tit);
        setLayout(new FlowLayout());
        add(t1=new TextField(10));
        add(t2=new TextField(10));
        t1.addTextListener(this);
        addWindowListener(this);
        addWindowListener(new WindowAdapter())
```

```
        {
            public void windowClosing(WindowEvent we)
            {
                System.exit(1);
            }
        }
    );
    setSize(300,300);
    setVisible(true);
}
public void textValueChanged(TextEvent et)
{
    t2.setText(t1.getText());
}
public void windowOpened(WindowEvent we){}
public void windowClosed(WindowEvent we){}
public void windowClosing(WindowEvent we){
    System.exit(1);
}
public void windowIconified(WindowEvent we){}
public void windowDeiconified(WindowEvent we){}
public void windowActivated(WindowEvent we){}
public void windowDeactivated(WindowEvent we){}

}
class myfr2_demo
{
    public static void main(String [] ar)
    {
        new myfr2("hi");
    }
}
```

131. Draw a Frame to use card Layout and border layout to change card on button click?

```
import java.awt.*;
import java.awt.event.*;
class myfr1 extends Frame implements ActionListener
{
    Button b1,b2,b3,b4,b5;
    TextField t1;
    Panel p,p1,p2,p3,p4;
    Choice list;
    CardLayout clo;
    public myfr1(String tit)
    {
        super(tit);
        b1=new Button("1");
        b2=new Button("2");
        b3=new Button("3");
        b4=new Button("4");
        b5=new Button("5");
        p=new Panel();
        p1=new Panel();
        p2=new Panel();
        p3=new Panel();
        p4=new Panel();
        t1=new TextField(10);
        clo=new CardLayout();
        p.setLayout(clo);
        p1.add(t1);
        p2.add(b1);
        p3.add(new Label("Third"));
        p4.add(new Label("fourth"));
        list=new Choice();
        list.addItem("dairy milk");
        list.addItem("temptation");
        p4.add(list);
    }
}
```

```

        p.add(p1,"first");
        p.add(p2,"sec");
        p.add(p3,"thir");
        p.add(p4,"four");
        add(p,"Center");
        add(b2,"South");
        add(b3,"East");
        add(b4,"West");
        add(b5,"North");
//  b1.addActionListener(ts);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
setSize(200,200);
setVisible(true);
}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==b5)
        clo.show(p,"first");
    else if(ae.getSource()==b2)
        clo.show(p,"sec");
    else if(ae.getSource()==b3)
        clo.show(p,"thir");
    else if(ae.getSource()==b4)
        clo.show(p,"four");
}
}
class myfr1_demo
{
public static void main(String [] rag)
{
    new myfr1("hi");
}
}

```

MULTITHREADING

Multithreading is a process of executing multiple threads simultaneously.

A multi-threaded program contains two or more parts that can run concurrently and each part can handle a different task at the same time.

Threads share a common memory area. They don't allocate separate memory area so, saves memory, and context-switching between the threads takes less time than process.

There are two ways to create a thread:

1. By extending Thread class
2. By implementing Runnable interface.

132. Write an applet to move ball across screen?

```
import java.applet.Applet;
import java.awt.*;
/*<applet code="thrap.class" height=200 width=400></applet>*/
public class thrap extends Applet implements Runnable
{
    Thread t;
    int x;
    public void init()
    {
        t=new Thread(this,"my thread");
        x=400;
        t.start();
    }

    public void run()
    {
        while(true){
            try{
                repaint();
                if(x<=0)
```

```

        x=400;
    else
        x=x-10;
    Thread.sleep(300);
}
catch(Exception ex){}
}
}
public void paint(Graphics g)
{
    g.setColor(Color.red);
    g.fillOval(x,20,50,50);
}
}

```

133. Write a Java applet to move ball alternate on screen?

```

import java.applet.Applet;
import java.awt.*;
/*<applet code="thrap1.class" height=200 width=400></applet>*/
public class thrap1 extends Applet implements Runnable
{
    Thread t;
    int x,y;
    public void init()
    {
        t=new Thread(this,"my thread");
        x=0;
        y=0;
        t.start();
    }

    public void run()
    {
        while(true){

```

```
        try{
            repaint();
            if(y==0)
            {
                if(x>=400)
                {
                    y=1;
                }
            }
            else
            {
                x=x+10;
            }
        }
        else if(y==1)
        {
            if(x<=0)
            {
                y=0;
            }
            else
            {
                x=x-10;
            }
        }
        Thread.sleep(300);
    }
    catch(Exception ex){}
}
}

public void paint(Graphics g)
{
    g.setColor(Color.red);
    g.fillOval(x,20,50,50);
}
}
```

134. Create a digital watch?

```
import javax.swing.*;
import java.awt.*;
```



```

import java.text.*;
import java.util.*;
public class DigitalWatch implements Runnable{
    JFrame f;
    Thread t=null;
    int hours=0, minutes=0, seconds=0;
    String timeString = "";
    JButton b;

    DigitalWatch(){
        f=newJFrame();

        t=newThread(this);
        t.start();

        b=newJButton();
        b.setBounds(100,100,100,50);

        f.add(b);
        f.setSize(300,400);
        f.setLayout(null);
        f.setVisible(true);
    }

    publicvoidrun()
    {
        try
        {
            while(true)
            {
                Calendarcal=Calendar.getInstance();
                hours=cal.get(Calendar.HOUR_OF_DAY);
                if(hours>12)hours-=12;
                minutes=cal.get(Calendar.MINUTE);
                seconds=cal.get(Calendar.SECOND);
            }
        }
    }
}

```

```
SimpleDateFormat formatter=newSimpleDateFormat
("hh:mm:ss");
Date date=cal.getTime();
timeString=formatter.format(date);

printTime();

t.sleep(1000);//interval given in milliseconds
}
}
catch(Exception e){}
}

public void printTime()
{
b.setText(timeString);
}

public static void main(String[] args)
{
new DigitalWatch();
}
}
```

135. Write a Java program to simulate traffic light system?

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class TrafficLight extends JFrame implements Runnable
{
    JButton red, green, yellow ;
    TrafficLight()
    {
        setTitle("TrafficLight");
    }
}
```

```

setSize(500,500) ;
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
red = new JButton() ;
yellow = new JButton() ;
green = new JButton() ;
setLayout(new GridLayout(3,1));
this.add(red); this.add(yellow); this.add(green) ;
Thread t = new Thread(this) ;
    t.start();
}
public void run()
{
    while(true)
    {
        Thread turnRed = new Thread(new TurnRed());
        turnRed.start() ;
        synchronized(turnRed)
        {
            try { turnRed.wait() ; } catch(InterruptedException e)
            {}
        }
        Thread turnYellow = new Thread(new TurnYellow());
        turnYellow.start() ;
        synchronized(turnYellow)
        {
            try { turnYellow.wait() ; } catch(InterruptedException
e) {}
        }
        Thread turnGreen = new Thread(new TurnGreen());
        turnGreen.start() ;
        synchronized(turnGreen)
        {
            try { turnGreen.wait() ; } catch(InterruptedException

```

```
e) {}  
    }  
    }  
}  
public static void main(String[] args)  
{  
    new TrafficLight().setVisible(true);  
}  
class TurnRed implements Runnable  
{  
    public void run()  
    {  
        synchronized(this) {  
            green.setBackground(Color.WHITE) ;  
            red.setBackground(Color.RED) ;  
            try { Thread.sleep(4000); } catch(InterruptedException
```

```
e) {}  
    notify(); }  
    }  
}  
class TurnYellow implements Runnable  
{  
    public void run()  
    {  
        synchronized(this) {  
            red.setBackground(Color.WHITE) ;  
            yellow.setBackground(Color.YELLOW) ;  
            try { Thread.sleep(5000); } catch(InterruptedException
```

```
e) {}  
    notify(); }  
    }  
}  
class TurnGreen implements Runnable
```

```

{
    public void run()
    {
        synchronized(this) {
            yellow.setBackground(Color.WHITE) ;
            green.setBackground(Color.GREEN) ;
            try { Thread.sleep(4000); } catch(InterruptedException
e) {}
                notify(); }
        }
    }
}

```

136. Write a program to start progress bar on clicking button?

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class progr extends JFrame implements Runnable, ActionListener
{
    JProgressBar jp;
    JButton b;
    Thread t;
    int i;
    public progr(String tit)
    {
        setLayout(new FlowLayout());
        i=0;
        b=new JButton("start");
        jp=new JProgressBar(0, 100);
        t=new Thread(this);
        getContentPane().add(b);
        getContentPane().add(jp);
    }
}

```

```
b.addActionListener(this);
setSize(200,200);
setVisible(true);
}
public void run()
{
while(true)
{
jp.setValue(jp.getValue()+i);
try{Thread.sleep(1200);}
catch(Exception er){}
i+=5;
}}
public void actionPerformed(ActionEvent ae)
{
    t.start();
}
```

```
public static void main(String [] arg)
{
    new progr("hi");
}

}
```

137. Write a program to print name in [] [] brackets using thread?

```
class callme
{
public void fun(String msg){
    System.out.print("[");
    try{
        Thread.sleep(100);
```

```

        System.out.print(msg);
    }catch(Exception ex){}
    System.out.println("]");
}
}
class call implements Runnable
{
    Thread t;
    callme c;
    String m;
    public call(String m,callme c)
    {
        this.m=m;
        this.c=c;
        t=new Thread(this);
        t.start();
    }
    public void run()
    {
        synchronized(c)
        {
            c.fun(m);
        }
    }
}
class calldemo
{
    public static void main(String [] ar)
    {
        callme c=new callme();
        call c1=new call("swati",c);
        call c2=new call("computers",c);
        try{c1.t.join(100);c2.t.join(100);}catch(Exception ex){}
    }
}

```

138. Write a program to change line position randomly?

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
/*<applet code="Game1.class" height=300 width=1000></applet>*/
public class Game1 extends Applet implements Runnable//,KeyListener
{
    int x,y,z;
    Thread t;
    Random rn;
    public void init()
    {
        x=500;
        y=120;
        t= new Thread(this);
        //add KeyListener(this);
        t.start();
    }
    public void paint(Graphics g)
    {
        g.drawLine(x,y,x+100,y);
    }
    public void run()
    {
        rn=new Random();
        while(true)
        {
            y=rn.nextInt(100);
            if(y<30)y+=20;
            if(x<=0)x=500;else x=x-50;
            repaint();
            try{Thread.sleep(1000);}catch(Exception ed){}
        }
    }
}
```


139. Write a program to create two threads and display numbering.

```
class threads1 extends Thread
{
    int i;
    public void run()
    {
        for(i=1;i<=10;i++)
        {
            System.out.println("One "+i);
            try{
                Thread.sleep(200);
            }
            catch(Exception e){}
        }
    }
}

class threads2 extends Thread
{
    int i;
    public void run()
    {
        for(i=1;i<=10;i++)
        {
            System.out.println("Two "+i);
            try{
                Thread.sleep(100);
            }
            catch(Exception e){}
        }
    }
}

class threads_demo
{
    public static void main(String [] ar)
```

```
{
    threads1 t=new threads1();
    threads2 t1=new threads2();
    t.start();
    t1.start();
}
}
```

140. Write a program to create two threads, set priority, name, and display their execution?

```
class thr extends Thread
{
    int i;
    public void run(){
        for(i=1;i<=10;i++)
        {
            try
            {
                System.out.println("Hello "+this.getName() +": "+i);
                Thread.sleep(1000);
            }
        }
        catch(InterruptedException ex){}
    }
}

class thr1 extends Thread
{
    int i;
    public void run(){
        for(i=100;i<=110;i++)
        {
            try
            {
                System.out.println("Hello "+this.getName() +": "+i);
                Thread.sleep(1000);
            }
        }
    }
}
```

```
    }  
    catch(InterruptedException ex){}  
    }  
}  
  
}  
public class thr_demo {  
    public static void main(String [] arg)  
    {  
        System.out.println(Thread.currentThread().getName());  
        thr t=new thr();  
        t.setName("first");  
        System.out.println("First running");  
        thr1 t1= new thr1();  
        t1.setName("Second");  
  
        t.setPriority(1);  
        t1.setPriority(7);  
        t1.start();  
        System.out.println("Second chance");  
        t.start();  
    }  
}
```

NETWORKING

Networking refers to the concept of connecting more than two devices to share resources

Java socket programming provides facility to share data between different computing devices.

The widely used Java networking terminologies are given as follows:

1. IP Address
2. Protocol
3. Port Number
4. MAC Address
5. Connection-oriented and connection-less protocol
6. Socket

1. IP Address

IP address is a unique number assigned to a node of a network e.g. 192.168.0.1 . It is composed of octets that range from 0 to 255. It has 5 classes A to E. It is a logical address that can be changed.

2. Protocol

A protocol is a set of rules , followed for communication.

TCP

FTP

Telnet

SMTP

POP etc.

3. Port Number

The port number is used to uniquely identify different applications. It acts as a communication endpoint between applications.

The port number is associated with the IP address for communication between two applications.

4. MAC Address

Media Access Control (MAC) Address is a unique identifier of **Network Interface Controller (NIC)**. A network node can have multiple NIC but each with unique MAC.

5. Connection-oriented and connection-less protocol

In connection-oriented protocol, acknowledgement is sent by the receiver. So, it is reliable but slow.

In connection-less protocol, acknowledgement is not sent by the receiver. So it is not reliable but fast.

6. Socket

A socket is an endpoint between two way communication.

The java.net package provides support for the two common network protocols:

TCP: Transmission Control Protocol (TCP), which allows for reliable communication between two applications. TCP is typically used over the Internet Protocol, which is referred to as TCP/IP.

UDP: User Datagram Protocol (UDP), a connection-less protocol that allows for packets of data to be transmitted between applications.

A datagram is an independent, self-contained message sent over the network whose

arrival, arrival time, and content are not guaranteed.

141. Write a chat program using UDP.

UDP Client:

```
import java.io.*;
import java.net.*;
```

```
class UDPCClient
```

```
{
```

```
public static DatagramSocket clientsocket;
```

```
public static DatagramPacket dp;
public static BufferedReader dis;
public static InetAddress ia;
public static byte buf[] = new byte[1024];
public static int cport = 789, sport = 790;
public static void main(String[] a) throws IOException
{
    clientsocket = new DatagramSocket(cport);
    dp = new DatagramPacket(buf, buf.length);
    dis = new BufferedReader(new
    InputStreamReader(System.in));
    ia = InetAddress.getLocalHost();
    System.out.println("Client is Running... Type 'STOP'

to Quit");
    while(true)
    {
        String str = new String(dis.readLine());
        buf = str.getBytes();
        if(str.equals("STOP"))
        {
            System.out.println("Terminated...");
            clientsocket.send(new
            DatagramPacket(buf, str.length(), ia,
            sport));
            break;
        }
        clientsocket.send(new DatagramPacket(buf,
        str.length(), ia, sport));
        clientsocket.receive(dp);
        String str2 = new String(dp.getData(), 0,
        dp.getLength());
        System.out.println("Server: " + str2);
    }
}
```

```
}
```

UDP Server:

```
import java.io.*;
import java.net.*;
class UDPServer
{
    public static DatagramSocket clientsocket;
    public static DatagramPacket dp;
    public static BufferedReader dis;
    public static InetAddress ia;
    public static byte buf[] = new byte[1024];
    public static int cport = 789,sport=790;
    public static void main(String[] a) throws IOException
    {
        serversocket = new DatagramSocket(sport);
        dp = new DatagramPacket(buf,buf.length);
        dis = new BufferedReader
            (new InputStreamReader(System.in));
        ia = InetAddress.getLocalHost();
        System.out.println("Server is Running...");
        while(true)
        {
            serversocket.receive(dp);
            String str = new String(dp.getData(), 0,
                dp.getLength());
            if(str.equals("STOP"))
            {
                System.out.println("Terminated...");
                break;
            }
            System.out.println("Client: " + str);
            String str1 = new String(dis.readLine());
            buf = str1.getBytes();
            serversocket.send(new
```

```
DatagramPacket(buf,str1.length(), ia, cport));  
}  
}  
}
```

142. Create a chat window using Server / ServerSocket.

```
//CLIENT  
import java.io.*;  
import java.net.*;  
import java.util.Scanner;  
import java.awt.*;  
import java.awt.event.*;  
import javax.swing.*;  
  
public class CClient extends JFrame implements ActionListener  
{  
    JTextField msg;  
    JTextArea txtarea;  
    byte[] sbfr,cbfr;  
    String senddata;  
    DatagramSocket client,server;  
    JButton button1;  
  
    public static void main(String[] args)  
    {  
        CClient obj=new CClient ();  
    }  
  
    public CClient ()  
    {  
        this.setSize(300,300);  
        this.setTitle("Client");  
        msg=new JTextField();  
        this.add(msg,BorderLayout.NORTH);  
        txtarea=new JTextArea();
```



```

this.add(txtarea, BorderLayout.CENTER);
button1=new JButton("SEND");
this.add(button1, BorderLayout.SOUTH);
button1.addActionListener(this);
this.setVisible(true);
sbfr=new byte[1024];
cbfr=new byte[1024];
setDefaultCloseOperation(EXIT_ON_CLOSE);
try
{
    client=new DatagramSocket();
    server=new DatagramSocket(9998);
    while(true)
    {
        DatagramPacket datapack=new
        DatagramPacket(sbfr,sbfr.length);
        server.receive(datapack);
        String msg=new String(datapack.getData());
        txtarea.append("\nServer:"+msg);
    }
}

catch(Exception e){}

}

```

```

public void actionPerformed(ActionEvent e)
{
    try
    {
        if(e.getActionCommand()=="SEND")
        {
            String message=msg.getText();
            cbfr=message.getBytes();

```

```
        DatagramPacket sendpack=new
        DatagramPacket(cbfr,cbfr.length,
        InetAddress.getLocalHost(),9999);
        client.send(sendpack);
        txtarea.append("\nMyself:"+message);
        msg.setText("");
    }
    }
    catch(Exception a){}
}

// SERVER
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

import java.io.*;

import java.net.*;
public class CServer extends JFrame implements ActionListener
{
    JTextField msg;
    JTextArea txtarea;
    byte[] sbfr,cbfr;
    DatagramSocket client,server;
    JButton button1;
    public static void main(String [] args)
    {
        CServer server=new CServer ();
    }
    public CServer ()
    {
        this.setSize(300,300);
        this.setTitle("Server");
    }
}
```

```

msg=new JTextField(100);
this.add(msg, BorderLayout.NORTH);
txtarea=new JTextArea();
this.add(txtarea, BorderLayout.CENTER);
button1=new JButton("SEND");
this.add(button1, BorderLayout.SOUTH);
button1.addActionListener(this);
this.setVisible(true);
sbfr=new byte[1024];
cbfr=new byte[1024];
//setDefaultCloseOperation(EXIT_ON_CLOSE);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
try
{
    client=new DatagramSocket();
    server=new DatagramSocket(9999);
    while(true)
    {
        DatagramPacket datapack=new
        DatagramPacket(cbfr,cbfr.length);
        server.receive(datapack);
        String msg=new String(datapack.getData());
        txtarea.append("\nClient:"+msg);
    }
}
catch(Exception e){}

}

public void actionPerformed(ActionEvent e)
{
    try
    {
        if(e.getActionCommand()=="SEND")
        {

```

```
        String message=msg.getText();
        sbfr=message.getBytes();
        DatagramPacket sendpack=new
        DatagramPacket(sbfr,sbfr.length,
        InetAddress.getLocalHost(),9998);
        client.send(sendpack);
        txtarea.append("\nMyself:"+message);
        msg.setText("");
    }
}
catch(Exception a){}
}
```

143. Write a program to encode URL.

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
public class MyUrlEncode
{
    public static void main(String a[])
    {
        try
        {
            System.out.println(URLEncoder.encode("String with spaces",
            "UTF-8"));
            System.out.println(URLEncoder.encode("special chars: &%*",
            "UTF-8"));
        }
        catch (UnsupportedEncodingException ex)
        {
            ex.printStackTrace();
        }
    }
}
```

144. Write a program to decode URL.

```
import java.io.UnsupportedEncodingException;
import java.net.URLDecoder;

public class MyURLDecode
{
    public static void main(String a[])
    {
        try
        {
            System.out.println(URLDecoder.decode("special+
            chars%3A+%26%25*+", "UTF-8"));
        }
        catch (UnsupportedEncodingException ex)
        {
            ex.printStackTrace();
        }
    }
}
```

145. Write a program to get URL properties in Java.

```
import java.net.MalformedURLException;
import java.net.URL;

public class UrlProperties
{
    public static void main(String a[])
    {
        try
        {
            String url =
            "http://www.sccomputers/java_examples/?query=ok#header";
            URL myUrl = new URL(url);
            System.out.println("Protocol: "+myUrl.getProtocol());
            System.out.println("Host: "+myUrl.getHost());
        }
    }
}
```

```
        System.out.println("Port: "+myUrl.getPort());
        System.out.println("Authority of the URL: "+myUrl.getAuthority());
        System.out.println("Query: "+myUrl.getQuery());
        System.out.println("Reference: "+myUrl.getRef());
    }
    catch (MalformedURLException ex)
    {
        ex.printStackTrace();
    }
}
```

146. Write a program to get the IP address of Local host in Java.

```
import java.net.InetAddress;
import java.net.UnknownHostException;

public class IpAddress
{
    public static void main(String a[])
    {
        try
        {
            InetAddress ipAddr = InetAddress.getLocalHost();
            System.out.println(ipAddr.getHostAddress());
        }
        catch (UnknownHostException ex)
        {
            ex.printStackTrace();
        }
    }
}
```

147. Write a Java program to check whether server is alive or dead.

```
import java.io.*;
import java.net.*;

public class Ping
{
    public static void main(String args[])
    {
        try
        {
            Socket t = new Socket(args[0], 7);
            DataInputStream dis = new DataInputStream(t.getInputStream());
            PrintStream ps = new PrintStream(t.getOutputStream());
            ps.println("Hello");
            String str = dis.readLine();
            if (str.equals("Hello"))
                System.out.println("Alive!");
            else
                System.out.println("Dead or echo port not responding");
            t.close();
        }
        catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

FILE HANDLING (IO PACKAGE)

The java.io package contains all the classes required for input and output operations.

We can perform **file handling in Java** by Java I/O API.

A stream can be defined as a sequence of data. There are two kinds of Streams –

InputStream – The InputStream is used to read data from a source.

OutputStream – The OutputStream is used for writing data to a destination.

148. Write a program to display the properties of existing file.

```
import java.io.*;
public class files {
    public static void main(String [] ar)throws Exception
    {
        File f=new File("Ios.java");
        System.out.println(f.getAbsolutePath());
        System.out.println(f.exists());
        System.out.println(f.canRead());
        System.out.println(f.isDirectory());
        System.out.println(f.isFile());

    }
}
```

149. Write a program to copy an existing file using bytestreams.

```
import java.io.*;
class file_in_out
{
    public static void main(String [] ar)throws Exception
    {
```



```
//Read from file
```

```
File f=new File("Excep.java");
//FileInputStream fin=new FileInputStream("Ios.java");
FileInputStream fin=new FileInputStream(f);

FileOutputStream fout=new FileOutputStream("copy.txt");
byte [] b=new byte[(int)fin.available()];

fin.read(b,0,b.length);
String s=new String(b);
System.out.println(s);
fin.close();

//Write into file
b=s.getBytes();
fout.write(b,0,b.length);

System.out.println("File created");
fout.close();
}
}
```

150. Example to implement pipe stream.

```
public static void main(String args[])throws Exception{
    PipedOutputStream pout=new PipedOutputStream();
    PipedInputStream pin=new PipedInputStream();
    pout.connect(pin);//connecting the streams
    Thread t1=new Thread(){
        public void run()
        {
            for(int i=65;i<=90;i++)
            { try{
                pout.write(i);
                Thread.sleep(1000); }
            }
```

```
        catch(Exception e){}
        }
    } };
Thread t2=new Thread(){
    public void run(){
        try{ for(int i=65;i<=90;i++)
            System.out.println(pin.read()); }
        catch(Exception e){}
        } };
    t1.start(); t2.start(); }
/*
public static void main(String args[])throws Exception
{
    FileOutputStream fout=new FileOutputStream("mfile.txt");
    PrintStream pout=new PrintStream(fout);
    pout.println(1900);
    pout.println("Hello Students");
    pout.println("Welcome to Class");
    pout.close();
    fout.close();
}*/
```

151. Write a program to read / write file using char stream.

```
import java.io.*;
class file_in_out
{
    public static void main(String [] ar)throws Exception
    {

        int c;
        String msg;
        FileReader fr=new FileReader("Ios.java");
        while((c=fr.read())!=-1)
        {
            System.out.print((char)c);
        }
    }
}
```

```

BufferedReader br =new BufferedReader(new InputStreamReader
(System.in));
System.out.println("Enter a message");
msg=br.readLine();
FileWriter fw=new FileWriter("copy1.txt",true);
/*   fw.write(msg);*/
fw.append(msg);
fw.close();
System.out.println("File Created");
}
}

```

152. Write a program to list the file and directories in given directory.

```

import java.io.File;
public class ReadDir
{
    public static void main(String[] args)
    {
        File file = null;
        String[] paths;

        try
        {
            // create new file object
            file = new File("/java");

            // array of files and directory
            paths = file.list();

            // for each name in the path array
            for(String path:paths)
            {
                // prints filename and directory name
                System.out.println(path);
            }
        }
    }
}

```

```
    } catch (Exception e)
    {
        // if any error occurs
        e.printStackTrace();
    }
}
```

153. Write a program to read data from file using `DataInputStream`.

```
import java.io.*;
public class DataStreamExample
{
    public static void main(String[] args) throws IOException
    {
        InputStream input = new FileInputStream("D:\\testing.txt");
        DataInputStream inst = new DataInputStream(input);

        int count = input.available();
        byte[] arr = new byte[count];
        inst.read(arr);
        for (byte bt : arr) {
            char k = (char) bt;
            System.out.println(k+"-");
        }
    }
}
```

Serialization

Serialization is the process to writing the state of an object into file.

java.io.Serializable interface

`Serializable` is a marker interface (has no data member and method).

It is used to “mark” Java classes so that objects of these classes may get certain capability. The Cloneable and Remote are also marker interfaces.

It must be implemented by the class whose object you want to save.

154. Write a student class and serialize its object into s.txt?

```
import java.io.*;
public class student implements Serializable
{
    public String name;
    int age;
    String address;
    float fee;

    public student(String n)
    {
        name=n;
    }
    public void show()
    {
        System.out.println(name);
    }
}
.....
import java.io.*;
class obj_demo
{
    public static void main(String [] ar)throws Exception
    {
        FileOutputStream fout=new FileOutputStream("s.txt");
        ObjectOutputStream ou=new ObjectOutputStream(fout);

        student s=new student("Swati");
        ou.writeObject(s); //object serializtion

        ou.close();
```

```
fout.close();
System.out.println("Record submitted in file");

//deserialization
FileInputStream fin=new FileInputStream("s.txt");
ObjectInputStream oin=new ObjectInputStream(fin);
student s1=(student)oin.readObject();
s1.show();
}
}
```

155. Write a program to read content of file line by line?

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class ReadLinesFromFile {

    public static void main(String a[]){
        BufferedReader br = null;
        String strLine = "";
        try {
            br = new BufferedReader( new FileReader("fileName"));
            while( (strLine = br.readLine()) != null){
                System.out.println(strLine);
            }
        } catch (FileNotFoundException e) {
            System.err.println("Unable to find the file: fileName");
        } catch (IOException e) {
            System.err.println("Unable to read the file: fileName");
        }
    }
}
```

156. Write a program to check whether a file is hidden or not?

```
import java.io.*;

public class IsHidden
{
    public static void main(String[] args)
    { // enter the file name here.
        // create file object.
        File file = new File("E:/hide.txt");

        // this will check is the file hidden or not.
        boolean blnHidden = file.isHidden();

        // return result in true or false condition.
        System.out.println("Is the file " + file.getPath() + " hidden ?: " +
            blnHidden);
    }
}
```

157. Write a program to read and print all files from zip files?

```
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.zip.ZipEntry;
import java.util.zip.ZipInputStream;

public class FindFileInZipFile
{
    public void printFileList(String filePath)
    {
        // initializing the objects.
        FileInputStream fis = null;
        ZipInputStream Zis = null;
```

```
ZipEntry zEntry = null;
try
{
    fis = new FileInputStream(filePath);
    Zis = new ZipInputStream(new BufferedInput
        Stream(fis));

    // this will search the files while end of the zip.
    while((zEntry = Zis.getNextEntry()) != null)
    {
        System.out.println(zEntry.getName());
    }
    Zis.close();
}
catch (FileNotFoundException e)
{
    e.printStackTrace();
}
catch (IOException e)
{
    e.printStackTrace();
}
}
//main function
public static void main(String a[])
{
    // creating object of the file.
    FindFileInZipFile fff = new FindFileInZipFile();
    System.out.println("Files in the Zip are : ");

    // enter the path of the zip file with name.
    fff.printFileList("D:/JAVA.zip");
}
}
```


JAVA ADVANCE

Swing
JApplet
JDBC
Servlet
JSP

Swing

Java Swing is a lightweight **Graphical User Interface** (GUI) toolkit that includes a rich set of widgets.

It is a part of **Java Foundation Classes (JFC)** that is used to create window-based applications.

It is built on the top of (**Abstract Windowing Toolkit**) AWT API and entirely written in Java.

153. Write a Java program to create a tabbed pane?

```
import javax.swing.*;
import java.awt.*;
class tab_demo extends JFrame
{
    JTabbedPane tb1;
    JPanel p2,p1;
    JToolBar tb;
    JTextArea tf;
    JButton n,s,sa,c1,p;
    public tab_demo(String tit)
    {
        setLayout(null);
        tb=new JToolBar();
        tf=new JTextArea(10,100);

        n=new JButton(new ImageIcon("new.jpg"));
```

```
c=new JButton(new ImageIcon("cut2.jpg"));
c1=new JButton(new ImageIcon("copy2.jpg"));
p=new JButton(new ImageIcon("paste.jpg"));
s=new JButton(new ImageIcon("save.jpg"));
sa=new JButton(new ImageIcon("saveas2.jpg"));
tb.setBounds(0,0,800,100);
tb.add(n);
tb.add(s);
tb.add(sa);
tb.add(c);
tb.add(c1);
tb.add(p);
//add(tf);
//getContentPane().add(tp);
getContentPane().add(tb);
tb1=new JTabbedPane();
p2=new JPanel();
p2.setLayout(null);
tf.setBounds(0,0,800,800);
p1=new JPanel();
p2.add(tf);
p1.add(new JButton("click"));
tb1.add(p2,"First");
tb1.add(p1,"Second");
tb1.setBounds(0,110,800,200);
getContentPane().add(tb1);
setSize(200,200);
setVisible(true);
setDefaultCloseOperation(EXIT_ON_CLOSE);

}
public static void main(String [] rag)
{
    new tab_demo("hi");
}
}
```

158. Write a JFrame, open enquiry form, registration form, fee entry form on click on menuItem.

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
class MainForm extends JFrame implements ActionListener
{
    JMenuBar mb;
    JMenu nw,up,help;
    JMenuItem enqf,regf,phn,addrs,fee,conus,courses,feeF;
    public MainForm(String tit)
    {
        super(tit);
        mb=new JMenuBar();
        nw=new JMenu("New");
        up=new JMenu("Update");
        help=new JMenu("Help");
        enqf=new JMenuItem("Enquiry Form") ;
        regf=new JMenuItem("Registration Form") ;
        phn=new JMenuItem("Phone") ;
        addrs=new JMenuItem("Address") ;
        fee=new JMenuItem("Fee") ;
        feeF=new JMenuItem("Fee Entry:");
        conus=new JMenuItem("Contact Us") ;
        courses=new JMenuItem("Courses") ;
        nw.add(enqf);
        nw.add(regf);
        up.add(phn);
        up.add(addrs);
        up.add(fee);
        help.add(conus);
        help.add(courses);
        help.add(feeF);
        mb.add(nw);
        mb.add(up);
```

```
mb.add(help);
    setJMenuBar(mb);
    enqf.addActionListener(this);
    regf.addActionListener(this);
    feeF.addActionListener(this);
setDefaultCloseOperation(EXIT_ON_CLOSE);
setLocation(300,100);
setVisible(true);
setSize(600,500);

}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==enqf)
    {
        new EnquiryF("Enquiry Form");
    }
    if(ae.getSource()==regf)
    {
        new RegF("Registration Form");
    }
    if(ae.getSource()==feeF)
    {
        new FeeF("Fee Entry Form");
    }
}
}

public class StudentInfoForm {
    public static void main(String[] arg)
    {
        new MainForm("Student Form");
    }
}

.....
//Enquiry Form
```

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import java.sql.*;
public class EnquiryF extends JFrame implements ActionListener,
ItemListener {
```

```
JLabel st_name,st_phone;
JTextField txt_name,txt_phone;
JButton b;
JPanel p;
JComboBox course;
JRadioButton m,f;
JCheckBox same;
JTextArea addr1,addr2,res;
public EnquiryF(String tit)
{
    super(tit);
    setLayout(new FlowLayout());
    st_name=new JLabel("Full Name");
    st_phone=new JLabel("Phone");
    txt_name=new JTextField(10);
    txt_phone=new JTextField(10);
    b=new JButton("submit");
    String[]cou=new String[]{"C","C++","JAVA",".NET","ANDROID",""
    PHP"};
    course=new JComboBox(cou);
    p=new JPanel();
    m=new JRadioButton("Male",true);
    f=new JRadioButton("Female");
    ButtonGroup bg=new ButtonGroup();
    bg.add(m);
    bg.add(f);

    addr1=new JTextArea(5,10);
```

```
        addr2=new JTextArea(5,10);
        same=new JCheckBox("Same as Above");
        res=new JTextArea(5,12);
        same.addItemListener(this);
        b.addActionListener(this);
m.addItemListener(this);
    /*
    setLayout(null);
    st_name.setBounds(10,50,100,20);
    txt_name.setBounds(130,50,100,20);
    st_phone.setBounds(10,90,100,20);
    txt_phone.setBounds(130,90,100,20);
    b.setBounds(40,140,100,30);
    */
    getContentPane().add(st_name);
    getContentPane().add(txt_name);
    getContentPane().add(st_phone);
    getContentPane().add(txt_phone);
    getContentPane().add(course);
    getContentPane().add(m);
    getContentPane().add(f);
    getContentPane().add(new JLabel("Temporary"));
    getContentPane().add(addr1);
    getContentPane().add(same);
    getContentPane().add(new JLabel("Permanent"));
    getContentPane().add(addr2);
    getContentPane().add(b);

    setDefaultCloseOperation(HIDE_ON_CLOSE);
    setLocation(300,100);
    setVisible(true);
    setSize(600,500);
}
public void itemStateChanged(ItemEvent ie)
{
```

```

if(ie.getSource()==same)
{
if(ie.getStateChange()==1)
{
addr2.setText(addr1.getText());
}

}
}
public void actionPerformed(ActionEvent ae)
{}
}

.....
//Registration Form

import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import java.sql.*;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
public class RegF extends JFrame implements ActionListener,
ItemListener {

JLabel st_name,st_phone,fn,em,Adhar_no,fee;
JTextField txt_name,txt_phone,txt_fn,txt_em,txt_adhar,txt_fee,txt_regf;
JButton b;
JPanel p;
JComboBox course;
JRadioButton m,f;
JCheckBox same;
JTextArea addr1,addr2,res;
public RegF(String tit)
{
    super(tit);

```

```
setLayout(new FlowLayout());
st_name=new JLabel("Full Name");
st_phone=new JLabel("Phone");
fn=new JLabel("Father's Name");
em=new JLabel("Email Id.");
Adhar_no=new JLabel("Aadhar No.");
txt_name=new JTextField(10);
txt_phone=new JTextField(10);
txt_fn=new JTextField(10);
txt_em=new JTextField(10);
txt_adhar=new JTextField(10);
fee=new JLabel("Fee");
txt_fee=new JTextField(10);
txt_regf=new JTextField(10);
b=new JButton("submit");
String [] cou=new String[]
{"C","C++","JAVA",".NET","ANDROID","PHP"};
course=new JComboBox(cou);
p=new JPanel();
m=new JRadioButton("Male",true);
f=new JRadioButton("Female");
ButtonGroup bg=new ButtonGroup();
bg.add(m);
bg.add(f);

addr1=new JTextArea(5,10);
addr2=new JTextArea(5,10);
same=new JCheckBox("Same as Above");
res=new JTextArea(5,12);
same.addItemListener(this);
b.addActionListener(this);
m.addItemListener(this);
/*
setLayout(null);
st_name.setBounds(10,50,100,20);
```



```

txt_name.setBounds(130,50,100,20);
st_phone.setBounds(10,90,100,20);
txt_phone.setBounds(130,90,100,20);
b.setBounds(40,140,100,30);
*/
getContentPane().add(st_name);
getContentPane().add(txt_name);
getContentPane().add(st_phone);
getContentPane().add(txt_phone);
getContentPane().add(course);
getContentPane().add(m);
getContentPane().add(f);
getContentPane().add(fn);
getContentPane().add(txt_fn);
getContentPane().add(em);
getContentPane().add(txt_em);
getContentPane().add(new JLabel("Temporary"));
getContentPane().add(addr1);
getContentPane().add(same);
getContentPane().add(new JLabel("Permanent"));
getContentPane().add(addr2);
getContentPane().add(Adhar_no);
getContentPane().add(txt_adhar);
getContentPane().add(fee);
getContentPane().add(txt_fee);
getContentPane().add(new JLabel("Registration Fee:"));
getContentPane().add(txt_regf);
getContentPane().add(b);

setDefaultCloseOperation(HIDE_ON_CLOSE);
setLocation(300,100);
setVisible(true);
setSize(600,500);
}
public void itemStateChanged(ItemEvent ie)

```

```
{
if(ie.getSource()==same)
{
if(ie.getStateChange()==1)
{
addr2.setText(addr1.getText());
}

}
}
public void actionPerformed(ActionEvent ae)
{}
}
.....
//Fee entry Form
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import java.sql.DriverManager;
import java.sql.*;
public class FeeF extends JFrame implements ActionListener,
FocusListener{
    JLabel adhar_no,total_fee,cre,bal,due_fee;
    JTextField txt_adhar,txt_total,txt_cre,txt_bal;
    JButton submit,search;
    Icon m;
    int cred;
    public FeeF(String tit)
    {
        super(tit);
        setLayout(new FlowLayout());
        m=new ImageIcon("1.jpg");
        adhar_no=new JLabel("Adhar No.");
        total_fee=new JLabel("Total Fee");
        cre=new JLabel("Credit");
```

```

        bal=new JLabel("Balance");
        txt_adhar=new JTextField(10);
        txt_total=new JTextField(10);
        txt_total.setEditable(false);
        txt_cre=new JTextField(10);
        txt_bal=new JTextField(10);
        txt_bal.setEditable(false);
        due_fee=new JLabel("");
        submit=new JButton("Submit");
        // search=new JButton(m);
        search=new JButton("Search");
        getContentPane().add(adhar_no);
        getContentPane().add(txt_adhar);
        getContentPane().add(search);
        getContentPane().add(total_fee);
        getContentPane().add(txt_total);
        getContentPane().add(new JLabel("Due Fee:"));
        getContentPane().add(due_fee);
        getContentPane().add(cre);
        getContentPane().add(txt_cre);
        getContentPane().add(bal);
        getContentPane().add(txt_bal);
        getContentPane().add(submit);
        txt_cre.addFocusListener(this);
        search.addActionListener(this);
        submit.addActionListener(this);
        setDefaultCloseOperation(HIDE_ON_CLOSE);
        setLocation(300,100);
        setVisible(true);
        setSize(600,500);
    }

    public void actionPerformed(ActionEvent ae)
    {}
}

```

159. Write a Java program to select Movie from JList and display ticket price, after that select gender and age and give discount on ticket. CHECK: For senior citizen male give 10% discount, for every female give 10% discount and if female is above 60 years give 20% discount on ticket price?

```
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*;
import javax.swing.event.*;
import javax.swing.event.ListSelectionListener;
class mvtest extends JFrame implements ActionListener,ListSelection
Listener
{
JList lst;
String [] ll=new String[]{"Avengers","Spy Kids","K3G","Super Seven"};
JRadioButton ml,fm;
JLabel age;
JTextField tr,agt,pay;
JButton sh,o;
ButtonGroup gen;
JScrollPane jsp;
int ag;
double pr,d,npr;

public mvtest(String tit)
{
super(tit);
setLayout(new FlowLayout());
lst=new JList(ll);
jsp=new JScrollPane(lst,      ScrollPaneConstants.VERTICAL_
SCROLLBAR_ALWAYS,      ScrollPaneConstants.HORIZONTAL_
SCROLLBAR_ALWAYS);
ml=new JRadioButton("Male",true);
fm=new JRadioButton("Female");
gen=new ButtonGroup();
```

```

gen.add(ml);
gen.add(fm);
tr=new JTextField(6);
tr.setEditable(false);
agt=new JTextField(3);
pay=new JTextField(6);
sh=new JButton("Show Ticket");
o=new JButton("Ok");

getContentPane().add(new JLabel("Select Movie"));
getContentPane().add(jsp);
getContentPane().add(new JLabel("Ticket Rate"));
getContentPane().add(tr);
getContentPane().add(new JLabel("Select Gender"));
getContentPane().add(ml);
getContentPane().add(fm);
getContentPane().add(new JLabel("Age"));
getContentPane().add(agt);
getContentPane().add(sh);
getContentPane().add(new JLabel("Pay :-"));
getContentPane().add(pay);
getContentPane().add(o);
lst.addListSelectionListener( this);
sh.addActionListener(this);
o.addActionListener(this);
setVisible(true);
setSize(350,400);
    setResizable(false);
setDefaultCloseOperation(EXIT_ON_CLOSE);
}
public void valueChanged(ListSelectionEvent e)
{
    int a=lst.getSelectedIndex();
    // String a=lst.getSelectedValue().toString();
    //JOptionPane.showMessageDialog(this, a);

```

```
        if(a==0)//ddlj
            pr=100;
        if(a==1)//dangal
            pr=150;
        if(a==2)//ki & ka
            pr=130;
        if(a==3)//sultan
            pr=180;
        tr.setText(pr+"");
    }
    public void actionPerformed(ActionEvent ae)
    {
        if(ae.getSource()==sh)
        {
            d=0;
            ag=Integer.parseInt(agt.getText().toString());
            if(ml.isSelected()==true)
            {
                if(ag>=60)
                {
                    d=pr*.10;
                    npr=pr-d;
                }
                else
                    npr=pr;
            }
            else {
                if(ag>=60)
                {
                    d=pr*.20;
                    npr=pr-d;
                }
                else
                {
                    d=pr*.10;
                    npr=pr-d;
                }
            }
        }
    }
}
```

```

        }
    }
    pay.setText(npr+""");
}
if(ae.getSource()==o)
{
    pay.setText("");
    agt.setText("");
    tr.setText("");
    JOptionPane.showMessageDialog
    (this,"Collect your ticket and pay "+npr+" rs. \n ThankYou ");
}
}
}
}

```

```

public class Test2
{
    public static void main(String [] arg)
    {
        new mvtest("Hi");
    }
}

```

160. Create a scrollable list?

```

import javax.swing.*;
import java.awt.*;
class first extends JFrame
{
    JButton b;
    JList lis;
    ImageIcon i;
    JScrollPane js;
    public first(String tit)
    {
        super(tit);
    }
}

```

```
i=new ImageIcon("12.jpg");
setLayout(new FlowLayout());
// getContentPane().add(b=new JButton(i));
String[]cho={"silk","nut","kit","kat","perk","5star","safari","rumraisin","
almonds"};
lis=new JList(cho);
js=new JScrollPane(lis, JScrollPaneConstants.VERTICAL_
SCROLLBAR_ALWAYS, JScrollPaneConstants.HORIZONTAL_
SCROLLBAR_ALWAYS);
getContentPane().add(js);
setVisible(true);
setSize(200,200);
setDefaultCloseOperation(EXIT_ON_CLOSE);
}
}
public class Firsr_Jframe {
    public static void main(String [] ar)
    {
        new first("hi");
    }
}
```

161. Write a program to change border on clicking button?

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.util.*;
class ch_border extends JFrame implements ActionListener {
    Border bor;
    FileDialog f;
    JButton b,b1,b2,b3,b4;
    JPanel p;

    public ch_border (String tit)
```



```

{
    super(tit);
    double n=(Math.random()*100);
    System.out.println(n);
    p=new JPanel();
    b=new JButton("click");
    b1=new JButton("click");
    b2=new JButton("click");
    b3=new JButton("click");
    b4=new JButton("click");
    p.add(b);
    getContentPane().add(p);
    getContentPane().add(b1,"North");
    getContentPane().add(b2,"South");
    getContentPane().add(b3,"East");
    getContentPane().add(b4,"West");
    b.addActionListener(this);
    b1.addActionListener(this);
    b2.addActionListener(this);
    b3.addActionListener(this);
    b4.addActionListener(this);
    setSize(200,200);
    setVisible(true);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
}

public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==b)
    {
        f=new FileDialog(this, "open file", FileDialog.LOAD);
        f.setSize(200,200);
        f.setVisible(true);
    }
    if(ae.getSource()==b1)
    {

```

```
        bor=BorderFactory.createLineBorder(Color.yellow, 5);
        p.setBorder(bor);
    }
    if(ae.getSource()==b2)
    {
        bor=BorderFactory.createBevelBorder
        (BevelBorder.RAISED, Color.green,
        Color.yellow, Color.blue, Color.pink);
        p.setBorder(bor);
    }
    if(ae.getSource()==b3)
    {
        bor=BorderFactory.createTitledBorder("hello");
        b.setBorder(bor);
    }
    if(ae.getSource()==b4)
    {
        bor=BorderFactory.createMatteBorder
        (10, 10, 10, 10, new ImageIcon("1.jpg"));
        b.setBorder(bor);
    }
}

class change_border{
    public static void main (String [] arg)
    {
        new ch_border("hi");
    }
}
```

162. Write a Java program to select an item and give order to buy, customer can select or deselect item before conforme.

Input name, phone number, address of customer, create checkboxes if the user click on check box he can give quantity, then check for total price after that charge GST for 10% and display final bill?

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class test extends JFrame implements ActionListener ,ItemListener
{
JLabel ta,cn,ph,add,si,pi,pp,pc,pcho;
JTextField tcn,tph,ti,tp,tc,tcho;
JTextArea tadd;
JCheckBox i,p,c,cho;
CheckboxGroup cbg;
JButton b,cb,ob;
FlowLayout flo;
JPanel jp;
int a,bb,l,d;
double tpri;
public test(String tit)
{
super(tit);
setLayout(flo=new FlowLayout());
cn=new JLabel("Name");
ph=new JLabel("Phone");
add=new JLabel("Please Enter Your Address ");
si=new JLabel("enter items & select order");
tcn=new JTextField(20);
tph=new JTextField(20);
ti=new JTextField("0",20);
tp=new JTextField("0",20);
tc=new JTextField("0",20);
tcho=new JTextField("0",20);
```

```
tadd=new JTextArea(5,20);
jp=new JPanel();
jp.setLayout(new GridLayout(5,3));
b=new JButton("Place Order");
i=new JCheckBox("Ice-Cream");
p=new JCheckBox("Pizza");
c=new JCheckBox("Coldrink");
cho=new JCheckBox("Chocolate");
cbg=new CheckboxGroup();
ti.setEditable(false);
tp.setEditable(false);
tc.setEditable(false);
tcho.setEditable(false);
add(cn);
add(tcn);
add(ph);
add(tph);
add(add);
add(tadd);
add(si);
jp.add(i);
ti.setEditable(false);
jp.add(ti);
jp.add(new JLabel("50Rs./PC"));
jp.add(p);
tp.setEditable(false);
jp.add(tp);
jp.add(new JLabel("60Rs./PC"));
jp.add(c);
tc.setEditable(false);
jp.add(tc);
jp.add(new JLabel("70Rs./PC"));
jp.add(cho);
tcho.setEditable(false);
jp.add(tcho);
```

```

jp.add(new JLabel("80Rs./PC"));
add(jp);
add(b);
add(pi=new JLabel("0"));
add(pp=new JLabel("0"));
add(pc=new JLabel("0"));
add(pcho=new JLabel("0"));
add(cb=new JButton("click to conform"));
add(ta=new JLabel("total price"));
add(ob=new JButton("OK"));
cb.addActionListener(this);
ob.addActionListener(this);
b.addActionListener(this);
i.addItemListener(this);
p.addItemListener(this);
c.addItemListener(this);
cho.addItemListener(this);

setDefaultCloseOperation(EXIT_ON_CLOSE);
setSize(800,400);
setVisible(true);
}

public void actionPerformed(ActionEvent e)
{

if(e.getSource()==b)
{a=bb=l=d=0;
a=Integer.parseInt(ti.getText());
a=a*50;
pi.setText("Ice-cream"+a);
bb=Integer.parseInt(tp.getText());
bb=bb*60;
pp.setText("PIZZAaaaa"+bb);
l=Integer.parseInt(tc.getText());
l=l*70;

```

```
pc.setText("coldrink"+l);
d=Integer.parseInt(tcho.getText());
d=d*80;
pcho.setText("chocolate"+d);

tpri=a+bb+l+d;
}
if(e.getSource()==cb)
{
    tpri=tpri+(tpri*(.10));
    ta.setText("Total Amount="+tpri);
}
if(e.getSource()==ob)
{
    JOptionPane.showMessageDialog(this,"visit again");
    //System.exit(0);
}
}
public void itemStateChanged(ItemEvent e)
{

if(i.isSelected()==true)
{
    ti.setEditable(true);
}
else
{
    ti.setEditable(false);
    a=0;
    ti.setText("0");
}
if(p.isSelected()==true)
{
    tp.setEditable(true);
}
```

```

else
{
    tp.setEditable(false);
    bb=0;
    tp.setText("0");
}
if(c.isSelected()==true)
{
    tc.setEditable(true);
}
else{
    tc.setEditable(false);
    l=0;
    tc.setText("0");
}
if(cho.isSelected()==true)
{
    tcho.setEditable(true);
}
else
{
    tcho.setEditable(false);
    d=0;
    tcho.setText("0");
}
}

public class test2
{
    public static void main(String [] asd)
    {
        new test("order placing");
    }
}

```

163. Write a program to create a tree of colors and chocolates?

```
import java.awt.Color;
import javax.swing.*;
import javax.swing.event.*;
import java.awt.event.*;
import javax.swing.tree.DefaultMutableTreeNode;
class myfr1 extends JFrame
{
    public myfr1(String tit)
    {
        super(tit);
        DefaultMutableTreeNode root=new DefaultMutableTreeNode("root");
        javax.swing.tree. DefaultMutableTreeNode col=new javax.swing.tree.
        DefaultMutableTreeNode("colour");
        DefaultMutableTreeNode  choco=new  DefaultMutableTreeNode
        ("chocolate");
        DefaultMutableTreeNode red=new DefaultMutableTreeNode("red");
        DefaultMutableTreeNode  gren=new  DefaultMutableTreeNode
        ("green");
        DefaultMutableTreeNode  dm=new  DefaultMutableTreeNode
        ("dairymilk");
        DefaultMutableTreeNodekit=newDefaultMutableTreeNode("kitkat");

        col.add(red);
        col.add(gren);
        choco.add(dm);
        choco.add(kit);
        root.add(col);
        root.add(choco);

        JTree tree=new JTree(root);
        JScrollPane js=new JScrollPane(tree,

        ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS,
        ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);
```



```

tree.setBorder(BorderFactory.createLineBorder(Color.yellow));
getContentPane().add(js);
setSize(400,400);
setDefaultCloseOperation(EXIT_ON_CLOSE);
setVisible(true);
}
}
public class trees {

    public static void main(String [] ar)throws Exception
    {
        new myfr1("hi");
    }
}

```

164. Write a program to create scrollable JList, JConbobox, JRadioButton.

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class myfr9 extends JFrame implements ActionListener
{
    JButton b,b1;
    JLabel l,mg;
    JList list;
    JComboBox comb;
    JRadioButton male,female;
    JScrollPane js;
    Icon img;

    myfr9(String tit)
    {
        super(tit);
        String[]s={"red","green","yellow","pink","black","blue","white","black"};
        img=new ImageIcon("shiva2.jpg");
        setLayout(new FlowLayout());
    }
}

```

```
list=new JList(s);
comb=new JComboBox(s);
male=new JRadioButton("Male",true);
female=new JRadioButton("Female",false);
js=new JScrollPane(list,ScrollPaneConstants.VERTICAL_
SCROLLBAR_ALWAYS,ScrollPaneConstants.HORIZONTAL_
SCROLLBAR_ALWAYS);
b=new JButton("ok");
b1=new JButton("Close");
mg=new JLabel(img );
getContentPane().add(mg);
getContentPane().add(js);
getContentPane().add(comb);
getContentPane().add(male);
getContentPane().add(female);
getContentPane().add(b);
getContentPane().add(b1);
b.addActionListener(this);
b1.addActionListener(this);
setDefaultCloseOperation(EXIT_ON_CLOSE);
setSize(400,400);
setVisible(true);
}
public void actionPerformed(ActionEvent ae)
{
if(ae.getSource()==b1)
    setVisible(false);
}
}
public class frm9 {
public static void main(String [] arg)
{
new myfr9("hi");
}
}
```

165. Write a program to create a JTable.

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.TreeNode;

class myfr12 extends JFrame
{
    JLabel nam;
    String [][] rows={{"1","swati"},"2","shweta"}};
    String []cols={"roll","name"};
    JTable tab;
    public myfr12(String tit,String r)
    {
        super(tit);

        setLayout(new FlowLayout());
        getContentPane().add(nam=new JLabel("Welcome "+r));
        //tree=new JTree();
        tab=new JTable(rows,cols);
        getContentPane().add(tab);
        setSize(200,200);
        setDefaultCloseOperation(HIDE_ON_CLOSE);
        setVisible(true);

    }
}

public class frm12
{
    public static void main(String [] as)
    {
        new myfr12("hi","ok");
    }
}
```

166. Write a program to create a JMenuBar and open JDialog on click on MenuItem?

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class myfr10 extends JFrame implements ActionListener
{
    JButton b;
    JMenuBar bar;
    JMenu file ,ed;
    JMenuItem nw,op,co,pas;
    JTextField name;
    JDialog d;
    public myfr10(String tit)
    {
        super(tit);
        bar=new JMenuBar();
        file=new JMenu("File");
        ed=new JMenu("Edit");
        nw=new JMenuItem("New");
        op=new JMenuItem("open");
        co=new JMenuItem("copy");
        pas=new JMenuItem("paste");
        name=new JTextField(10);
        nw.addActionListener(this);
        op.addActionListener(this);
        file.add(nw);
        file.add(op);
        ed.add(co);
        ed.add(pas);
        bar.add(file);
        bar.add(ed);
        setJMenuBar(bar);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(1000,1000);
        setLocation(100,100);
    }
}
```

```

        setVisible(true);
    }
    public void actionPerformed(ActionEvent ae )
    {
        if(ae.getSource()==op)
        //...
        if(ae.getSource()==nw)
        {
            d=new JDialog(this, "hello");
            d.add(new JLabel("welcome user"));
            b=new JButton("Close me");
            d.add(b,"South");
            b.addActionListener(this);
            d.setSize(200,200);
            d.setLocation(200,200);
            d.setVisible(true);
        }
        if(ae.getSource()==b)
            d.setVisible(false);
    }
}
public class frm10 {
    public static void main(String [] arg)
    {
        new myfr10("hi");
    }
}

```

167. Write a program to show tooltip on button.

```

import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
class myfr11 extends JFrame implements ActionListener
{
    public myfr11(String tit)
    {

```

```
        super(tit);
        setLayout(new FlowLayout());
        JButton btn = new JButton("Button");
        btn.setToolTipText("A button component");
        getContentPane().add(btn);
    }
}
public class frm11 {
    public static void main(String [] as)
    {
        new myfr11("hi");
    }
}
```

168. Write a program to add slider on JPanel.

```
import javax.swing.*;
public class SliderExample extends JFrame
{
    public SliderExample() {
        JSlider slider = new JSlider(JSlider.HORIZONTAL, 0, 50, 25);
        slider.setMinorTickSpacing(2);
        slider.setMajorTickSpacing(10);
        slider.setPaintTicks(true);
        slider.setPaintLabels(true);

        JPanel panel=new JPanel();
        panel.add(slider);
        add(panel);
    }
    public static void main(String s[])
    {
        SliderExample frame=new SliderExample();
        frame.pack();
        frame.setVisible(true);
    }
}
```

JAPPLET

JApplet can have all the controls of swing. The JApplet class extends the Applet class.

169. Write a JApplet, on button click “ welcome message “ should display on textbox?

```
import java.applet.*;
import javax.swing.*;
import java.awt.event.*;
public class EventJApplet extends JApplet implements ActionListener{
    JButton b;
    JTextField tf;
    public void init()
    {
        tf=new JTextField();
        tf.setBounds(30,40,150,20);

        b=new JButton("Click");
        b.setBounds(80,150,70,40);

        add(b);add(tf);
        b.addActionListener(this);

        setLayout(null);
    }

    public void actionPerformed(ActionEvent e)
    {
        tf.setText("Welcome");
    }
}
.....
<html>
```

```
<body>
<applet code="EventJApplet.class" width="300" height="300">
</applet>
</body>
</html>
```

170. Write an applet to change image on button click.

```
import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;
/*<applet code="img.class" Height=700 Width=500></applet>*/
public class img extends Applet implements ActionListener
{
    Button b;
    String s;
    int d=0;
    Image m[];
    public void init()
    {
        m=new Image[3];

        m[0]=getImage(getDocumentBase(),"a.gif");
        m[1]=getImage(getDocumentBase(),"b.gif");
        m[2]=getImage(getDocumentBase(),"c.gif");
        b=new Button("click");
        b.addActionListener(this);
        add(b);
    }
    public void paint(Graphics g)
    {
        if(d==1)
            g.drawImage(m[0],30,40,this);
        if(d==2)
            g.drawImage(m[1],30,40,this);
        if(d==3)
```



```

g.drawImage(m[2],30,40,this);
}
public void actionPerformed(ActionEvent ae)
{
s=ae.getActionCommand();
if(s.equals("click"))
{d++;
if(d>3)
{d=1;}
}
repaint();
}
}

```

171. Write a program to draw line, circle, rectangle according to user input.

```

import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;

/*<applet code="draw.class" Height=500 Width=500></applet>*/
public class draw extends Applet implements ActionListener
{
String s;
int c;
TextField t;
TextField t1;
public void init()
{

t=new TextField(10);

add(t);
Button b1;
b1=new Button("Draw");
b1.addActionListener(this);

```

```
    add(b1);
  }
  public void paint(Graphics g)
  {
    if(c==1)
    {
      g.drawRect(20,20,20,20);
    }
    if(c==3)
    {
      g.drawOval(20,20,20,20);
    }
    if(c==2)
    {
      g.drawLine(20,20,40,40);
    }

  }
  public void actionPerformed(ActionEvent ae)
  {
    s=t.getText();
    if(s.equals ("rectangle"))
    {
      c=1;
    }
    if(s.equals ("line"))
    {
      c=2;
    }
    if(s.equals ("circle"))
    {
      c=3;
    }
    repaint();
  }
}
```

172. Write an applet to display selected item from list box (single as well as multiple), enter guest name, bill number, and display bill?

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
```

```
/* <applet code="selecteditem.class" height="500" width="500">
```

```
</applet> */
```

```
public class selecteditem extends Applet implements
```

```
ItemListener,ActionListener
```

```
{
    TextField t,t1;
    List l1;
    Button b;
    String p,q,r;
    int c;
    int d;
    Label l,l2;
    public void init()
    {
        l=new Label("Guest");
        l2=new Label("Bill number");
        t=new TextField(10);
        t1=new TextField(10);
        add(l);
        add(t);
        add(l2);
        add(t1);
        b=new Button("order");
        b.addActionListener(this);
        add(b);
        l1=new List(2,true);
```

```
l1.add("pizza");
l1.add("burger");
l1.add("hotdog");
l1.add("footlong");
l1.add("donuts");
add(l1);
l1.addItemListener(this);
}
public void itemStateChanged(ItemEvent e)
{
    repaint();
}
public void actionPerformed(ActionEvent ae)
{
    System.out.println("\007");
    r=t1.getText();
    q=t.getText();
    p=ae.getActionCommand();
    if(p.equals("order"))
    {
        c=1;
        repaint();
    }
}
public void paint(Graphics g)
{
    if(c==1)
    {
        g.drawString("Guest is"+q,150,430);
        g.drawString("Bill no. is"+r,150,440);
        g.drawString("your final bill is Rupees"+d,150,450);
        g.drawString("Have a nice day!",150,460);
    }
    int i=0;

    String val[];
```

```
String data="You have selected items";
val=l1.getSelectedItems();
int len=val.length;
while(len!=i)
{
    data=data+" "+val[i]+" ";
    String s="your expected bill is Rupees";
    int j=0;
    if(i==0)
    {j=j+10;
    g.drawString(s+j,150,170);
    d=j;
    }

    if(i==1)
    {j=j+20;
    g.drawString(s+j,150,180);
    d=j;
    }
    if(i==2)
    {j=j+30;
    g.drawString(s+j,150,190);
    d=j;
    }

    if(i==3)
    {j=j+40;
    g.drawString(s+j,150,200);
    d=j;
    }
    if(i==4)
    {j=j+50;
    g.drawString(s+j,150,210);
    d=j;
    }
```

```
i++;  
g.drawString(data,150,160);  
}  
  
}  
}
```

173. Write a program to implement cardLayout on mouseClicked.

```
import java.applet.Applet;  
import java.awt.*;  
import java.awt.event.*;  
/*<applet code="card.class" Height=700 Width=500></applet>*/  
public class card extends Applet implements MouseListener  
{  
    String s;  
    CardLayout cl=new CardLayout();  
    Panel p,p1,p2;  
    public void init()  
    {  
        addMouseListener(this);  
        setLayout(cl);  
        p=new Panel();  
        p1=new Panel();  
        p2=new Panel();  
        p.add(p1,"chocolate");  
        p.add(p2,"cream");  
        add(p);  
    }  
    public void mouseClicked(MouseEvent me)  
    {  
        cl.show(p,"ice cream");  
        cl.show(p,"chocolate");  
    }  
    public void mouseEntered(MouseEvent me)  
    {
```

```

}
public void mouseExited(MouseEvent me)
{
}
public void mouseReleased(MouseEvent me)
{
}
public void mousePressed(MouseEvent me)
{
}
}

```

174. Write a program to check the mouse position on mouseMove.

```

import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
/*<applet code="pos.class" Height=700 Width=500></applet>*/
public class pos extends Applet implements MouseMotionListener
{
    TextField t;
    public void init()
    {
        t=new TextField(10);
        add(t);
        addMouseMotionListener(this);
    }
    public void mouseMoved(MouseEvent me)
    {
        int x=me.getX();
        int y=me.getY();
        t.setText(x+" ":"+y);
    }
    public void mouseDragged(MouseEvent me)
    {}
}

```

JDBC

Java Database Connectivity (**JDBC**) is an **application programming interface (API)** which allows the programmer to connect and interact with databases.

It provides methods to query and update data in the database through update statements like SQL's CREATE, UPDATE, DELETE, and INSERT, and query statements such as SELECT.

Additionally, JDBC can run stored procedures.

Just like Java, JDBC is compatible with many platforms like Unix and MAC OS.

Before JDBC, ODBC API was the database API to connect and execute query with the database.

But, ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured).

That is why Java has defined its own API (JDBC API) that uses JDBC drivers (written in Java language).

The JDBC API uses Java standard classes and interfaces to connect to databases.

In order to use JDBC to connect Java applications to a specific database server, a JDBC driver that supports the JDBC API for that database server is required.

JDBC Driver:

JDBC driver implementations vary because of the wide variety of operating systems and hardware platforms in which Java operates.

JDBC Driver is a software component that enables Java application to interact with the database. There are 4 types of JDBC drivers:

1. JDBC-ODBC bridge driver
2. Native-API driver (partially Java driver)
3. Network Protocol driver (fully Java driver)
4. Thin driver (fully Java driver)

Type 1: JDBC-ODBC Bridge Driver

In a Type 1 driver, a JDBC bridge is used to access ODBC drivers installed on each client machine. Using ODBC, requires configuring on your system, a **Data Source Name (DSN)** that represents the target database.

When Java first came out, this was a useful driver because most databases only supported ODBC access but now this type of driver is recommended only for experimental use or when no other alternative is available.

Open Database Connectivity (ODBC) is an open standard API for accessing a database.

By using ODBC statements in a program, you can access files in a number of different databases, including Access, dBase, DB2, Excel, and Text. In addition to the ODBC software, a separate module or driver is needed for each database to be accessed.

The main of ODBC programming support is Microsoft.

Type 2: JDBC-Native API

In a Type 2 driver, JDBC API calls are converted into native C/C++ API calls, which are unique to the database.

These drivers are typically provided by the database vendors and used in the same manner as the JDBC-ODBC Bridge. The vendor-specific driver must be installed on each client machine.

The Native API driver uses the client-side libraries of the database. The driver converts JDBC method calls into native calls of the database API. It is not written entirely in Java.

Type 3: . Network Protocol driver / JDBC-Net pure Java

In a Type 3 driver, a three-tier approach is used to access databases. The JDBC clients use standard network sockets to communicate with a middleware application server.

The socket information is then translated by the middleware application server into the call format required by the DBMS, and forwarded to the database server.

Type 4: . Fully Java Driver

This is the highest performance driver available for the database and is usually provided by the vendor itself.

This kind of driver is extremely flexible, you don't need to install special software on the client or server. Further, these drivers can be downloaded dynamically.

Steps to connect your application to database:

- ❖ Import JDBC Packages
- ❖ Register the driver class
- ❖ Creating connection
- ❖ Creating statement
- ❖ Executing queries
- ❖ Closing connection

Connectivity With Ms-Access:

- Create DSN

```
String url="jdbc:odbc:mydsn";
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection c=DriverManager.getConnection(url);
Statement st=c.createStatement();
```

Connectivity with Oracle

- Firstly, search the ojdbc14.jar file then go to JRE/lib/ext folder and paste the jar file here.
- `Class.forName("oracle.jdbc.driver.OracleDriver");`

```
//step2 create the connection object
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
```

```
//step3 create the statement object
Statement stmt=con.createStatement();
```

Connectivity With MY-SQL

```
Class.forName("com.mysql.jdbc.Driver");
```

```
Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/database","root","root");
```

```
// root is username and password
```

```
Statement stmt=con.createStatement();
```

175. Create a JFrame with two text boxes for user name and password, and do DBMS oracle connectivity and create buttons for signup, sign in, update password, delete record?

```
import java.sql.*;
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import java.util.logging.Level;
import java.util.logging.Logger;
public class dbms_example extends JFrame implements ActionListener
{
    Connection con;
    Statement st;
    PreparedStatement ps;
    JTextField ut,pt;
    JButton b1,b2,b3,b4,b5;
    ResultSet rs;
    JLabel res;
    public dbms_example (String tit)
    {
        super(tit);
        setLayout(new FlowLayout());
        getContentPane().add(new JLabel("Enter user name"));
        getContentPane().add(ut=new JTextField(10));
```

```
getContentPane().add(new JLabel("Enter password"));
getContentPane().add(pt=new JTextField(10));
getContentPane().add(b1=new JButton("SignIn"));
getContentPane().add(b2=new JButton("SignUp"));
getContentPane().add(b3=new JButton("Update"));
getContentPane().add(b4=new JButton("Delete"));
getContentPane().add(b5=new JButton("forgot password"));
getContentPane().add(res=new JLabel("Result"));
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);

setSize(400,500);
setDefaultCloseOperation(EXIT_ON_CLOSE);
setVisible(true);
}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==b1)
    {
        conn();
        try
        {
            st=con.createStatement();
            rs=st.executeQuery("select * from sub_login where user_name='"+ut.
            getText()+"' and user_password='"+pt.getText()+"'");
            if(rs.next())
            {
                res.setText("Welcome user");
            }
            else
            {
                res.setText("Get lost..");
            }
        }
    }
}
```

```

        }
    } catch (SQLException er)
    {
        System.out.println(er.toString());
    }
}
if(ae.getSource()==b2)
{
    conn();
    try
    {
        ps=con.prepareStatement("insert into sub_login values(?,?)");
        ps.setString(1,ut.getText());
        ps.setString(2,pt.getText());
        ps.executeUpdate();
        res.setText("Record Submitted");
    }
    catch(Exception er){System.out.println(er.toString());}
}
if(ae.getSource()==b3)
{
    conn();
    try
    {
        ps=con.prepareStatement("update sub_login set user_password=?
        where
        user_name=?");
        ps.setString(2,ut.getText());
        ps.setString(1,pt.getText());
        ps.executeUpdate();
        res.setText("updated");
    }
    catch(Exception er){System.out.println(er.toString());}
}
if(ae.getSource()==b4)

```

```
{
    conn();
    try
    {
        ps=con.prepareStatement("delete from sub_login where user_name=?
        and user_password=?");
        ps.setString(1,ut.getText());
        ps.setString(2,pt.getText());
        int r=ps.executeUpdate();
        res.setText(r+"record delete");
    }
    catch(Exception er){System.out.println(er.toString());}
}
    if(ae.getSource()==b5)
    {
        conn();
        try
        {
            ps=con.prepareStatement("select user_password from sub_login
            where user_name=?");
            ps.setString(1,ut.getText());
            rs=ps.executeQuery();
            if(rs.next())
            {
                res.setText(rs.getString(1));
            }
            else
                res.setText("record not found");
        }
        catch(Exception er){System.out.println(er.toString());}
    }

}
public void conn()
{
```

```

    try{
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","root");
    }
    catch(Exception er){System.out.println(er.toString());}
}
public static void main(String[] args)
{
    new dbms_example ("hi");
}
}

```

176. Create a student information for with database connectivity for new enquiry and new registration?

```

import javax.swing.*.*;
import java.awt.event.*;
import java.awt.*.*;
class MainForm extends JFrame implements ActionListener
{
    JMenuBar mb;
    JMenu nw,up,help;
    JMenuItem enqf,regf,phn,addrs,fee,conus,courses,feeF;
    public MainForm(String tit)
    {
        super(tit);
        mb=new JMenuBar();
        nw=new JMenu("New");
        up=new JMenu("Update");
        help=new JMenu("Help");
        enqf=new JMenuItem("Enquiry Form") ;
        regf=new JMenuItem("Registration Form") ;
        phn=new JMenuItem("Phone") ;
        addrs=new JMenuItem("Address") ;
    }
}

```

```
fee=new JMenuItem("Fee") ;
feeF=new JMenuItem("Fee Entry:");
conus=new JMenuItem("Contact Us") ;
courses=new JMenuItem("Courses") ;
nw.add(enqf);
nw.add(regf);
up.add(phn);
up.add(addr);
up.add(fee);
help.add(conus);
help.add(courses);
help.add(feeF);
mb.add(nw);
mb.add(up);
mb.add(help);
    setJMenuBar(mb);
    enqf.addActionListener(this);
    regf.addActionListener(this);
    feeF.addActionListener(this);
setDefaultCloseOperation(EXIT_ON_CLOSE);
setLocation(300,100);
setVisible(true);
setSize(600,500);

}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==enqf)
    {
        new EnquiryF("Enquiry Form");
    }
    if(ae.getSource()==regf)
    {
        new RegF("Registration Form");
    }
}
```



```

if(ae.getSource()==feeF)
{
    new FeeF("Fee Entry Form");
}
}
}
}
public class StudentInfoForm
{
    public static void main(String[] arg)
    {
        new MainForm("Student Form");
    }
}
.....
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import java.sql.*;

```

```

public class EnquiryF extends JFrame implements ActionListener,
ItemListener
{
    Connection con;
    PreparedStatement ps;
    JLabel st_name,st_phone;
    JTextField txt_name,txt_phone;
    JButton b;
    JPanel p;
    JComboBox course;
    JRadioButton m,f;
    JCheckBox same;
    JTextArea addr1,addr2,res;
    public EnquiryF(String tit)
    {
        super(tit);

```

```
setLayout(new FlowLayout());
st_name=new JLabel("Full Name");
st_phone=new JLabel("Phone");
txt_name=new JTextField(10);
txt_phone=new JTextField(10);
b=new JButton("submit");
String[] cou=new String[]{"C","C++","JAVA",".NET","ANDROID","PHP"};
course=new JComboBox(cou);
p=new JPanel();
m=new JRadioButton("Male",true);
f=new JRadioButton("Female");
ButtonGroup bg=new ButtonGroup();
bg.add(m);
bg.add(f);

addr1=new JTextArea(5,10);
addr2=new JTextArea(5,10);
same=new JCheckBox("Same as Above");
res=new JTextArea(5,12);
same.addItemListener(this);
b.addActionListener(this);
m.addItemListener(this);
/*
setLayout(null);
st_name.setBounds(10,50,100,20);
txt_name.setBounds(130,50,100,20);
st_phone.setBounds(10,90,100,20);
txt_phone.setBounds(130,90,100,20);
b.setBounds(40,140,100,30);
*/
getContentPane().add(st_name);
getContentPane().add(txt_name);
getContentPane().add(st_phone);
getContentPane().add(txt_phone);
```

```
getContentPane().add(course);
getContentPane().add(m);
getContentPane().add(f);
getContentPane().add(new JLabel("Temporary"));
getContentPane().add(addr1);
getContentPane().add(same);
getContentPane().add(new JLabel("Permanent"));
getContentPane().add(addr2);
getContentPane().add(b);

setDefaultCloseOperation(HIDE_ON_CLOSE);
setLocation(300,100);
setVisible(true);
setSize(600,500);
}
public void conn()
{
    try
    {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con=DriverManager.getConnection("jdbc:oracle:thin:@
localhost:1521:XE","system","root");
    }
    catch(Exception ex){}
}
public void itemStateChanged(ItemEvent ie)
{
    if(ie.getSource()==same)
    {
        if(ie.getStateChange()==1)
        {
            addr2.setText(addr1.getText());
        }
    }
}
```

```
    }
    public void actionPerformed(ActionEvent ae)
    { String g="";
    if(ae.getSource()==b)
    {

        if(m.isSelected()==true)
            g=m.getText();
        else
            g=f.getText();
        res.setText("");
        res.setText(txt_name.getText()+","+txt_phone.getText()+","+course.
        getSelectedItem()+","+g+","+addr1.getText()+","+addr2.getText());

        try
        {
            conn();
            ps=con.prepareStatement("insert into table_enq values(?,?,?,?,?)");
            ps.setString(1,txt_name.getText());
            ps.setInt(2,Integer.parseInt(txt_phone.getText()));
            ps.setString(3,course.getSelectedItem().toString());
            ps.setString(4,g);
            ps.setString(5, addr1.getText());
            ps.setString(6,addr2.getText());
            ps.executeUpdate();
        }
        catch(Exception ex)
        {
            System.out.println(ex.toString());
        }
    }
}
}
}
}
.....
import javax.swing.*;
```

```
import java.awt.event.*;
import java.awt.*;
import java.sql.*;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class RegF extends JFrame implements ActionListener,
ItemListener {

    Connection con;
    PreparedStatement ps;
    JLabel st_name,st_phone,fn,em,Adhar_no,fee;
    JTextField txt_name,txt_phone,txt_fn,txt_em,txt_adhar,txt_fee,txt_regf;
    JButton b;
    JPanel p;
    JComboBox course;
    JRadioButton m,f;
    JCheckBox same;
    JTextArea addr1,addr2,res;
    public RegF(String tit)
    {
        super(tit);
        setLayout(new FlowLayout());
        st_name=new JLabel("Full Name");
        st_phone=new JLabel("Phone");
        fn=new JLabel("Father's Name");
        em=new JLabel("Email Id.");
        Adhar_no=new JLabel("Aadhar No.");
        txt_name=new JTextField(10);
        txt_phone=new JTextField(10);
        txt_fn=new JTextField(10);
        txt_em=new JTextField(10);
        txt_adhar=new JTextField(10);
        fee=new JLabel("Fee");
        txt_fee=new JTextField(10);
```

```
txt_regf=new JTextField(10);
b=new JButton("submit");
String[]cou=new String[]{"C","C++","JAVA",".NET","ANDROID","
PHP"};
course=new JComboBox(cou);
p=new JPanel();
m=new JRadioButton("Male",true);
f=new JRadioButton("Female");
ButtonGroup bg=new ButtonGroup();
bg.add(m);
bg.add(f);

addr1=new JTextArea(5,10);
addr2=new JTextArea(5,10);
same=new JCheckBox("Same as Above");
res=new JTextArea(5,12);
same.addItemListener(this);
b.addActionListener(this);
m.addItemListener(this);
/*
setLayout(null);
st_name.setBounds(10,50,100,20);
txt_name.setBounds(130,50,100,20);
st_phone.setBounds(10,90,100,20);
txt_phone.setBounds(130,90,100,20);
b.setBounds(40,140,100,30);
*/
getContentPane().add(st_name);
getContentPane().add(txt_name);
getContentPane().add(st_phone);
getContentPane().add(txt_phone);
getContentPane().add(course);
getContentPane().add(m);
getContentPane().add(f);
getContentPane().add(fn);
```

```

    getContentPane().add(txt_fn);
    getContentPane().add(em);
    getContentPane().add(txt_em);
    getContentPane().add(new JLabel("Temporary"));
    getContentPane().add(addr1);
    getContentPane().add(same);
    getContentPane().add(new JLabel("Permanent"));
    getContentPane().add(addr2);
    getContentPane().add(Adhar_no);
    getContentPane().add(txt_adhar);
    getContentPane().add(fee);
    getContentPane().add(txt_fee);
    getContentPane().add(new JLabel("Registration Fee:"));
    getContentPane().add(txt_regf);
    getContentPane().add(b);

    setDefaultCloseOperation(HIDE_ON_CLOSE);
    setLocation(300,100);
    setVisible(true);
    setSize(600,500);
}
public void conn()
{
    try
    {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con=DriverManager.getConnection("jdbc:oracle:thin:@
localhost:1521:XE","system","root");
    }
    catch(Exception ex){}
}
public void itemStateChanged(ItemEvent ie)
{
    if(ie.getSource()==same)

```

```
{
if(ie.getStateChange()==1)
{
addr2.setText(addr1.getText());
}

}
}
public void actionPerformed(ActionEvent ae)
{
if(ae.getSource()==b)
{
String g="";
if(m.isSelected()==true)
g=m.getText();
else
g=f.getText();
try{
conn();

ps=con.prepareStatement("insert      into      table_Reg      values
(?,?,?,?,?,?,?,?)");
ps.setString(1,txt_name.getText());
ps.setString(2,txt_phone.getText());
ps.setString(3,course.getSelectedItem().toString());
ps.setString(4,g);
ps.setString(5,txt_fn.getText());
ps.setString(6,txt_em.getText());

ps.setString(7,addr1.getText());
ps.setString(8,addr2.getText());
ps.setString(9,txt_adhar.getText());
ps.setInt(10,Integer.parseInt(txt_fee.getText()));
ps.executeUpdate();
ps.close();
```



```

ps=con.prepareStatement("insert into table_fees values(?,?,?,?)");
ps.setString(1,txt_adhar.getText());
ps.setInt(2,Integer.parseInt(txt_fee.getText()));
ps.setInt(3,Integer.parseInt(txt_regf.getText()));
int due=Integer.parseInt(txt_fee.getText())-Integer.parseInt(txt_regf.
getText());
ps.setInt(4,due);
ps.executeUpdate();
}
catch(Exception ex){ex.printStackTrace();}
}
}
}

```

177. Write a JFrame with userName and Password field, new user can register but if user name is already registered, same user name is not acceptable, user can update password, delete account and login also?

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
class dbdemo extends JFrame implements ActionListener
{
    JLabel lnm,lpwd,lres;
    JTextField tnm,tpwd;
    JButton b,b1,b2,b3,b4;
    Connection con;
    Statement st;
    PreparedStatement ps;
    ResultSet rs;
    String p,u;
    public dbdemo(String tit)
    {
        setLayout(new GridLayout(5,2));
        add(lnm=new JLabel("User Name"));

```

```
add(tnm=new JTextField(20));
add(lpwd=new JLabel("Password"));
add(tpwd=new JTextField(20));
add(b=new JButton("Register"));
add(b1=new JButton("Login"));
add(b2=new JButton("Update Pasword"));
add(b3=new JButton("Delete Record"));
add(b4=new JButton("Show All"));
add(lres=new JLabel("Result"));

b.addActionListener(this);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);

setSize(350,200);
setResizable(false);
setVisible(true);
setDefaultCloseOperation(EXIT_ON_CLOSE);
}
public void conn()
{
    try{
        Class.forName("oracle.jdbc.driver.OracleDriver");
        con=DriverManager.getConnection
("jdbc:oracle:thin:@localhost:1521:XE","system","root");
    }catch(Exception ex){}
}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==b) //register
    {
        try {
```

```

        conn();
    /*
        st=con.createStatement();
        st.executeUpdate("insert into users_tab values(
"+tnm.getText()+"','"+tpwd.getText()+"')");
    */
    ps=con.prepareStatement
    ("select * from users_tab where uname=?");
    ps.setString(1, tnm.getText());
    rs=ps.executeQuery();
    if(rs.next())
    {
        JOptionPane.showMessageDialog(this, "User already
        registered, user name is not available");
        rs.close();
    }
    else{
        ps=con.prepareStatement
    ("insert into users_tab values(?,?)");
        ps.setString(1,tnm.getText());
        ps.setString(2,tpwd.getText());
        ps.executeUpdate();
        lres.setText("Data Submitted");
    }
    } catch (Exception ex) {}
    }
    if(ae.getSource()==b1) //register
    {
        try{
            conn();
            ps=con.prepareStatement
    ("select * from users_tab
            where uname=? and pwd=?");

```

```
        ps.setString(1,tnm.getText());
        ps.setString(2,tpwd.getText());
        rs=ps.executeQuery();
        if(rs.next())
            lres.setText("WelCome User");
        else
            lres.setText("Invalid Try again");
        rs.close();
    }catch(Exception ex){}
    }
    if(ae.getSource()==b2) //update
    {
        try{
            conn();
            ps=con.prepareStatement("update users_tab set pwd=?
                                   and where uname=?");
            ps.setString(2,tnm.getText());
            ps.setString(1,tpwd.getText());
            ps.executeUpdate();
            lres.setText("Record Update");
        }catch(Exception ex){}
    }
    if(ae.getSource()==b3) //delete
    {
        try{
            conn();
            ps=con.prepareStatement("delete from users_tab where
uname=? and pwd=?");
            ps.setString(1,tnm.getText());
            ps.setString(2,tpwd.getText());
            ps.executeUpdate();
            lres.setText("Record Deleted");
        }catch(Exception ex){}
    }
}
```

```

if(ae.getSource()==b4) //show
{
    try{
        conn();
        ps=con.prepareStatement("select * from users_tab");
        rs=ps.executeQuery();
        while(rs.next())
        {
            System.out.print(rs.getString(1));
            System.out.println("\t"+rs.getString(2));
        }
        rs.close();
    }catch(Exception ex){}
}
}

```

```

public class dbs_demo {
    public static void main(String [] poo)
    {
        new dbdemo("User_Account");
    }
}

```

178. Write a JDBC program to select item name from database in JList and user can choose items from JList and display into another JList, user can also remove selected items?

```

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*;
import java.sql.*;
class shop extends JFrame implements ActionListener
{
    JButton ch,ex;

```

```
JList dlist,clist;
Connection con;
Statement st;
PreparedStatement ps;
ResultSet rs;
String []db=new String[100];
String []cl;
int i=0;
JScrollPane js,js1;
DefaultListModel<String> model=new DefaultListModel<String>();
DefaultListModel<String> model1=new DefaultListModel<String>();
public shop(String tit)
{
    super(tit);
    setLayout(new FlowLayout());
    try{ conn();
        ps=con.prepareStatement("select * from pra_tb");
        rs=ps.executeQuery();
        while(rs.next())
        {
            db[i++]=rs.getString(1);
            dlist=new JList(db);
        }

        rs.close();}catch(Exception ex){}
    add(js=new JScrollPane(dlist,

ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS,
ScrollPan

eConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED));
    add(ch=new JButton(">>"));
    add(ex=new JButton("<<"));
    clist=new JList();
    add(js1=new JScrollPane(clist,
```

```
ScrollPaneConstants.VERTICAL_SCROLLBAR_
ALWAYS,ScrollPane
```

```
eConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED));
```

```
        ch.addActionListener(this);
        ex.addActionListener(this);
        setVisible(true);
        setSize(400,300);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
    public void conn()
    {
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            con=DriverManager.getConnection
            ("jdbc:oracle:thin:@localhost:1521:XE","system","root");
        }catch(Exception ex){}
    }
```

```
    public void actionPerformed(ActionEvent ae){
        if(ae.getSource()==ch){
            model.addElement(dlist.getSelectedValue().toString());
            model1.addElement(dlist.getSelectedValue().toString());
            clist.setModel(model);
        }
        if(ae.getSource()==ex)
        {
            i=clist.getSelectedIndex();
            model1.remove(i);
            clist.setModel(model1);
        }
    }
}
```

```
public class admins {  
    public static void main(String [] arg)  
    {  
        new shop("Select Item");  
    }  
}
```

179. Write a JDBC program to display students data in ascending order of FirstName?

```
import java.sql.*;  
  
public class JDBCExample {  
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";  
    static final String DB_URL = "jdbc:mysql://localhost/STUDENTS";  
  
    static final String USER = "username";  
    static final String PASS = "password";  
  
    public static void main(String[] args) {  
        Connection conn = null;  
        Statement stmt = null;  
  
        try{  
            Class.forName("com.mysql.jdbc.Driver");  
  
            System.out.println("Connecting to a selected database...");  
            conn = DriverManager.getConnection(DB_URL, USER, PASS);  
            System.out.println("Connected database successfully...");  
  
            stmt = conn.createStatement();  
  
            System.out.println("Fetching records in ascending order...");  
            String sql = "SELECT id, first, last, age FROM Registration" +  
                " ORDER BY first ASC";
```



```
ResultSet rs = stmt.executeQuery(sql);

while(rs.next()){
    int id = rs.getInt("id");
    int age = rs.getInt("age");
    String first = rs.getString("first");
    String last = rs.getString("last");

    System.out.print("ID: " + id);
    System.out.print(", Age: " + age);
    System.out.print(", First: " + first);
    System.out.println(", Last: " + last);
}

System.out.println("Fetching records in descending order...");
sql = "SELECT id, first, last, age FROM Registration" +
      " ORDER BY first DESC";
rs = stmt.executeQuery(sql);

while(rs.next()){
    //Retrieve by column name
    int id = rs.getInt("id");
    int age = rs.getInt("age");
    String first = rs.getString("first");
    String last = rs.getString("last");

    //Display values
    System.out.print("ID: " + id);
    System.out.print(", Age: " + age);
    System.out.print(", First: " + first);
    System.out.println(", Last: " + last);
}
rs.close();
} catch (SQLException se){
```

```
        se.printStackTrace();
    }catch(Exception e){
        e.printStackTrace();
    }finally{
        try{
            if(stmt!=null)
                conn.close();
        }catch(SQLException se){
        }
        try{
            if(conn!=null)
                conn.close();
        }catch(SQLException se){
            se.printStackTrace();
        }
    }
    System.out.println("Goodbye!");
}}
```

180. Write a JDBC application to call stored procedure.

```
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class MySimpleCallableStatement {

    public static void main(String a[]){

        Connection con = null;
        CallableStatement callSt = null;
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            con=DriverManager.getConnection("jdbc:oracle:thin:@
localhost:1521:XE","system","root");
```

```

        callSt = con.prepareCall("{call myprocedure(?,?)}");
        callSt.setInt(1,200);
        callSt.setDouble(2, 3000);
        callSt.execute();
        System.out.println("Executed stored procedure!!!");
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally{
        try{
            if(callSt != null) callSt.close();
            if(con != null) con.close();
        } catch (Exception ex){}
    }
}
}

```

181. Write a JDBC application to get column properties using resultsetmeta data?

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;

public class ResultSetMetadata_use{

    public static void main(String a[]){

        Connection con = null;
        Statement st = null;
        ResultSet rs = null;
        try {

```

```
Class.forName("oracle.jdbc.driver.OracleDriver");
con=DriverManager.getConnection
("jdbc:oracle:thin:@localhost:1521:XE","system","root");
    st = con.createStatement();
    rs = st.executeQuery("select * from emp");
    ResultSetMetaData rsmd = rs.getMetaData();
    int columnCount = rsmd.getColumnCount();
    for(int i=0;i<=columnCount;i++){
        System.out.println(rsmd.getColumnName(i));
        System.out.println(rsmd.getColumnType(i));
    }
} catch (ClassNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} finally{
    try{
        if(rs != null) rs.close();
        if(st != null) st.close();
        if(con != null) con.close();
    } catch(Exception ex){}
}
}
```

182. Write a program to insert image (picture) into database?

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
```

```
import java.sql.SQLException;

public class Insert_image {

    public static void main(String a[]){

        Connection con = null;
        PreparedStatement ps = null;
        InputStream is = null;
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            con=DriverManager.getConnection
            ("jdbc:oracle:thin:@localhost:1521:XE","system","root");

            ps = con.prepareCall("insert into student_profile values (?,?)");
            ps.setInt(1, 101);
            is = new FileInputStream(new File("Student_img.jpg"));
            ps.setBinaryStream(2, is);
            int count = ps.executeUpdate();
            System.out.println("Count: "+count);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } finally{
            try{
                if(is != null) is.close();
                if(ps != null) ps.close();
                if(con != null) con.close();
            } catch (Exception ex){}
        }
    }
}
```

183. Write a program to read image from database.

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Read_image {

    public static void main(String a[]){

        Connection con = null;
        Statement st = null;
        ResultSet rs = null;
        InputStream is = null;
        OutputStream os = null;
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            con=DriverManager.getConnection
            ("jdbc:oracle:thin:@localhost:1521:XE","system","root");
            st = con.createStatement();
            rs = st.executeQuery
            ("select student_img from student_profile where id=101");
            if(rs.next()){
                is = rs.getBinaryStream(1);
            }
            is = new FileInputStream(new File("Student_img.jpg"));
            os = new FileOutputStream("std_img.jpg");
```

```

        byte[] content = new byte[1024];
        int size = 0;
        while((size = is.read(content)) != -1){
            os.write(content, 0, size);
        }
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally{
        try{
            if(is != null) is.close();
            if(os != null) os.close();
            if(st != null) st.close();
            if(con != null) con.close();
        } catch (Exception ex){}
    }
}
}

```

184. Write a program to create progress monitor using JProgressBar.

```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;
import javax.swing.ProgressMonitor;
import javax.swing.SwingUtilities;
import javax.swing.Timer;
import javax.swing.UIManager;

```

```
public class Progress_Monitor extends JFrame implements ActionListener {

    static ProgressMonitor pbar;

    static int counter = 0;

    public Progress_Monitor () {
        super("Progress Monitor Demo");
        setSize(250, 100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        pbar = new ProgressMonitor(null, "Monitoring Progress",
            "Initializing ...", 0, 100);

        Timer timer = new Timer(500, this);
        timer.start();
        setVisible(true);
    }

    public static void main(String args[])
    {
        UIManager.put("ProgressMonitor.progressText", "This is progress?");
        UIManager.put("OptionPane.cancelButtonText", "Close Me");
        new ProgressMonitorExample();
    }

    public void actionPerformed(ActionEvent e)
    {
        // Invoked by the timer every half second. Simply place
        // the progress monitor update on the event queue.
        SwingUtilities.invokeLater(new Update());
    }
}
```



```

class Update implements Runnable {
    public void run() {
        if (pbar.isCanceled()) {
            pbar.close();
            System.exit(1);
        }
        pbar.setProgress(counter);
        pbar.setNote("Operation is " + counter + "% complete");
        counter += 2;
    }
}

```

185. Write a JDBC application to insert data of friends, using SQL connectivity.

```

import java.sql.*;

public class FriendsData {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/FRIENDS";

    static final String USER = "username";
    static final String PASS = "password";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try{
            Class.forName("com.mysql.jdbc.Driver");

            System.out.println("Connecting to a selected database...");
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
            System.out.println("Connected database successfully...");

            System.out.println("Inserting records into the table...");

```

```
stmt = conn.createStatement();

String sql = "INSERT INTO Friend_List " +
             "VALUES ('Swati','1-6-2000','9829440773)";
stmt.executeUpdate(sql);
sql = "INSERT INTO Friend_List " +
      "VALUES (('Shweta','30-1-1990','9829440773)";
stmt.executeUpdate(sql);
sql = "INSERT INTO Friend_List " +
      "VALUES (('Juhi','1-6-1989','9829440773)";
stmt.executeUpdate(sql);
sql = "INSERT INTO Friend_List " +
      "VALUES (('Sweety','15-12-2014','9829440773)";
stmt.executeUpdate(sql);
System.out.println("Inserted records into the table...");

}catch(SQLException se){
    //Handle errors for JDBC
    se.printStackTrace();
}catch(Exception e){
    e.printStackTrace();
}finally{
    try{
        if(stmt!=null)
            conn.close();
    }catch(SQLException se){
    }    try{
        if(conn!=null)
            conn.close();
    }catch(SQLException se){
        se.printStackTrace();
    }
} System.out.println("Goodbye!");
}
}
```

186. Write a JDBC application to display data from database in ascending/descending order?

```
import java.sql.*;

public class JDBC_Sorting {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/STUDENTS";

    static final String USER = "username";
    static final String PASS = "password";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try{
            Class.forName("com.mysql.jdbc.Driver");

            System.out.println("Connecting to a selected database...");
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
            System.out.println("Connected database successfully...");

            System.out.println("Creating statement...");
            stmt = conn.createStatement();

            // Extract records in ascending order by first name.
            System.out.println ("Fetching records in ascending order...");
            String sql = "SELECT id, first, last, age FROM Registration" +
                " ORDER BY first ASC";
            ResultSet rs = stmt.executeQuery(sql);

            while(rs.next()){
                //Retrieve by column name
                int id = rs.getInt("id");
```

```
int age = rs.getInt("age");
String first = rs.getString("first");
String last = rs.getString("last");

//Display values
System.out.print("ID: " + id);
System.out.print(", Age: " + age);
System.out.print(", First: " + first);
System.out.println(", Last: " + last);
}

// Extract records in descending order by first name.
System.out.println("Fetching records in descending order...");
sql = "SELECT id, first, last, age FROM Registration" +
      " ORDER BY first DESC";
rs = stmt.executeQuery(sql);

while(rs.next()){
    //Retrieve by column name
    int id = rs.getInt("id");
    int age = rs.getInt("age");
    String first = rs.getString("first");
    String last = rs.getString("last");

    System.out.print("ID: " + id);
    System.out.print(", Age: " + age);
    System.out.print(", First: " + first);
    System.out.println(", Last: " + last);
}
rs.close();
}catch(SQLException se){
    se.printStackTrace();
}catch(Exception e){
    e.printStackTrace();
}
```

```

}finally{

    try{
        if(stmt!=null)
            conn.close();
    }catch(SQLException se){
    }
    try{
        if(conn!=null)
            conn.close();
    }catch(SQLException se){
        se.printStackTrace();
    }
}
System.out.println("Goodbye!");
}
}

```

187. Write a JDBC application to delete table.

```

import java.sql.*;

public class JDBC_DeleteTable {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost/STUDENTS";

    static final String USER = "username";
    static final String PASS = "password";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;

        try{
            Class.forName("com.mysql.jdbc.Driver");

```

```
System.out.println("Connecting to a selected database...");
conn = DriverManager.getConnection(DB_URL, USER, PASS);
System.out.println("Connected database successfully...");

System.out.println("Deleting table in given database...");
stmt = conn.createStatement();

String sql = "DROP TABLE REGISTRATION ";

stmt.executeUpdate(sql);
System.out.println("Table deleted in given database...");
}catch(SQLException se){

    se.printStackTrace();
}catch(Exception e){

    e.printStackTrace();
}finally{

    try{
        if(stmt!=null)
            conn.close();
    }catch(SQLException se){
    }
    try{
        if(conn!=null)
            conn.close();
    }catch(SQLException se){
        se.printStackTrace();
    }
}
System.out.println("Goodbye!");
}
}
```

188. Fetch data from SQL database table and display in JTable?

```
import java.awt.*;
import java.sql.*;
import java.util.*;
import javax.swing.*;
import javax.swing.table.*;

public class Fetch_Table extends JFrame
{
    public Fetch_Table ()
    {
        ArrayList columnNames = new ArrayList();
        ArrayList data = new ArrayList();

        String url = "jdbc:mysql://localhost:3306/yourdb";
        String userid = "root";
        String password = "sesame";
        String sql = "SELECT * FROM animals";

        try (Connection connection = DriverManager.getConnection
            ( url, userid, password );
            Statement stmt = connection.createStatement();
            ResultSet rs = stmt.executeQuery( sql ))
        {
            ResultSetMetaData md = rs.getMetaData();
            int columns = md.getColumnCount();

            // Get column names
            for (int i = 1; i <= columns; i++)
            {
                columnNames.add( md.getColumnName(i) );
            }

            // Get row data
```

```
while (rs.next())
{
    ArrayList row = new ArrayList(columns);

    for (int i = 1; i <= columns; i++)
    {
        row.add( rs.getObject(i) );
    }

    data.add( row );
}
catch (SQLException e)
{
    System.out.println( e.getMessage() );
}

Vector columnNamesVector = new Vector();
Vector dataVector = new Vector();

for (int i = 0; i < data.size(); i++)
{
    ArrayList subArray = (ArrayList)data.get(i);
    Vector subVector = new Vector();
    for (int j = 0; j < subArray.size(); j++)
    {
        subVector.add(subArray.get(j));
    }
    dataVector.add(subVector);
}

for (int i = 0; i < columnNames.size(); i++ )
    columnNamesVector.add(columnNames.get(i));

// Create table with database data
```



```

JTable table = new JTable(dataVector, columnNamesVector)
{
    public Class getColumnClass(int column)
    {
        for (int row = 0; row < getRowCount(); row++)
        {
            Object o = getValueAt(row, column);

            if (o != null)
            {
                return o.getClass();
            }
        }

        return Object.class;
    }
};

JScrollPane scrollPane = new JScrollPane( table );
getContentPane().add( scrollPane );

JPanel buttonPanel = new JPanel();
getContentPane().add( buttonPanel, BorderLayout.SOUTH );
}

public static void main(String[] args)
{
    Fetch_Table frame = new Fetch_Table ();
    frame.setDefaultCloseOperation( EXIT_ON_CLOSE );
    frame.pack();
    frame.setVisible(true);
}
}

```

189. Write a program to draw shapes after selecting from Menu.

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;

public class Paintpad extends JFrame
{
    JMenuBar mmain;
    JMenu shapes,color;
    JMenuItem rect,circle,arc;
    int x1=10,x2=100,y1=10,y2=100,sh=0;
    public Paintpad(String title)
    {

        mmain=new JMenuBar();
        shapes=new JMenu("Shapes");
        color=new JMenu("Color");
        rect=new JMenuItem("Rectangle");
        circle=new JMenuItem("Circle");
        arc=new JMenuItem("Arc");
        shapes.add(rect);
        shapes.add(circle);
        shapes.add(arc);
        mmain.add(shapes);
        mmain.add(color);
        setJMenuBar(mmain);
        setTitle(title);
        setSize(300,300);
        setVisible(true);
        setLayout(new FlowLayout());
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        rect.addActionListener(new userAction());
```

```

circle.addActionListener(new userAction());
arc.addActionListener(new userAction());
addMouseListener(new MouseListener() {
    @Override
    public void mouseClicked(MouseEvent e) {}
    @Override
    public void mousePressed(MouseEvent e) {
        x1=e.getX();
        y1=e.getY();

    }
    @Override
    public void mouseReleased(MouseEvent e) {
        x2=e.getX();
        y2=e.getY();
        drawshape(sh);

    }
    @Override
    public void mouseEntered(MouseEvent e) {}
    @Override
    public void mouseExited(MouseEvent e) {}));
}
public class userAction implements ActionListener
{

    @Override
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==rect)
        {
            sh=1;
        }
    }
}

```

```
        if(e.getSource()==circle)
        {
            sh=2;
        }
        if(e.getSource()==arc)
        {
            sh=3;
        }
    }

}

public void drawshape(int i)
{
    //sh=0;
    Graphics g=getGraphics();
    g.setColor(Color.red);
    if(i==1)
    {
        if(x1>x2){x1=x1+x2;x2=x1-x2;x1=x1-x2;}
        if(y1>y2){y1=y1+y2;y2=y1-y2;y1=y1-y2;}
        g.drawRect(x1, y1, x2-x1, y2-y1);
    }
    if(i==2)
    {
        if(x1>x2){x1=x1+x2;x2=x1-x2;x1=x1-x2;}
        if(y1>y2){y1=y1+y2;y2=y1-y2;y1=y1-y2;}
        g.drawOval(x1, y1, x2-x1, y2-y1);
    }
    if(i==3)
    {
        if(x1>x2){x1=x1+x2;x2=x1-x2;x1=x1-x2;}
        if(y1>y2){y1=y1+y2;y2=y1-y2;y1=y1-y2;}
        g.drawArc((x1+x2)/2, (y1+y2)/2, (x2-x1), (y2-y1), 0, 180);
    }
}
```

```
    }  
}
```

```
public static void main(String ar[])  
{  
    new Paintpad("Drawing");  
}  
}
```

SERVLET

After the Web began to be used for delivering services, service providers recognized the need for dynamic content.

Developers investigated using the server platform for the same purpose. **Common Gateway Interface (CGI)** server-side scripts were the main technology used to generate dynamic content.

But, CGI scripting technology had many shortcomings, including platform dependence and lack of scalability.

To address these limitations, Java Servlet technology was created as a portable way to provide dynamic, user-oriented content.

Servlet can be described as:

- Servlet is a technology, used to create web application.
- Servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases.
- Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request.
- A servlet is a Java programming language class used to extend the capabilities of servers that host applications accessed by means of a request-response programming model.
- Servlets are platform-independent because they are written in Java.
- Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any type of requests.

CGI technology enables the web server to call an external program and pass HTTP request information to the external program to process the request. For each request, it starts a new process.

A number of Web Servers that support servlets are available in the market. Some web servers are freely downloadable and Tomcat is one of them.

The `javax.servlet` package contains many interfaces and classes that are used by the servlet or web container.

The `javax.servlet.http` package contains interfaces and classes that are responsible for http requests only.

Servlet Life cycle:

1. Servlet class is loaded.
2. Servlet instance is created.
3. init method is invoked.
4. service method is invoked.
5. destroy method is invoked.
 - The servlet is initialized by calling the init() method.
 - The servlet calls service() method to process a client's request.
 - The servlet is terminated by calling the destroy() method.

Type of Servlet:

1. Generic Servlet: GenericServlet class can handle any type of request so it is protocol-independent.
2. HttpServlet: The HttpServlet class extends the GenericServlet class and implements Serializable interface.

It provides http specific methods such as doGet, doPost, doHead, doTrace etc.

190. Write a servlet to print "hello" message.

```
import java.io.*;
import javax.servlet.*;

public class First implements Servlet{
    ServletConfig config=null;

    public void init(ServletConfig config){
        this.config=config;
        System.out.println("servlet is initialized");
    }

    public void service(ServletRequest req,ServletResponse res)
        throws IOException,ServletException{
```

```
res.setContentType("text/html");

PrintWriter out=res.getWriter();
out.print("<html><body>");
out.print("<b>hello Swati Computers</b>");
out.print("</body></html>");

}
public void destroy(){System.out.println("servlet is destroyed");}
public ServletConfig getServletConfig(){return config;}
public String getServletInfo(){return "";}

}
```

191. Write a HTTPServlet to print Hello message.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Second extends HttpServlet {

    private String message;

    public void init() throws ServletException {
        message = "Hello Swati Computers";
    }

    public void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {

        // Set response content type
        response.setContentType("text/html");

        // Actual logic goes here.
        PrintWriter out = response.getWriter();
```



```

        out.println("<h1>" + message + "</h1>");
    }

    public void destroy() {
        // do nothing.
    }
}

```

192. Write a servlet to input user name and print welcome message (in NetBeans).

a.html

```

<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
  </head>
  <body>
    <form action=".\\servlet1" method="post">
      Enter name<input type="text" name="nam">
      <input type="submit">
    </form>
  </body>
</html>

```

Servlet1

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.eclipse.jdt.internal.compiler.util.Util;

```

```

public class servlet1 extends HttpServlet {

```

```
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    out.write("Welcome user "+request.getParameter("nam"));
}

}
```

194. Write a servlet to check user name and password, if login is invalid go back to HTML page to login again or print welcome message to user?

Form1.html

```
<html>
<head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
</head>
<body>
    <form action=".\\servlet1" method="post">
        Enter name<input type="text" name="nam"><br>
        Enter password<input type="password" name="pwd">
        <input type="Login">
    </form>
</body>
</html>
```

Servlet1

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class servlet1 extends HttpServlet {

    String n,p;
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
//out.write("Welcome user
        "+request.getParameter("nam"));
        n=request.getParameter("nam");
        p=request.getParameter("pwd");
        RequestDispatcher      rd=request.getRequestDispatcher
        ("servlet2");
        if(n.equals("swati") && p.equals("computers"))
        {
            rd.forward(request, response);
        }
        else
        {
            out.println("Invalid Login");
            rd=request.getRequestDispatcher("form1.html");
            rd.include(request, response);
        }
    }
}

```

Servlet2

```

import java.io.IOException;
import java.io.PrintWriter;

```

```
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class servlet2 extends HttpServlet {
    protected void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        res.setContentType("text/html");
        PrintWriter pw=res.getWriter();
        pw.write("welcome user " + req.getParameter("nam"));
    }
}
```

1. Write a servlet to redirect user to a given website.

Form.html

```
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
  </head>
  <body>
    <form action=".\\servlet1" method="post">
      Enter Website you want to open:<br>
      https://www.<input type="text" name="nam"><br>
      <input type="submit" value="Send">
    </form>
  </body>
</html>
```

Servlet1

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
```

```

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class servlet1 extends HttpServlet {

    String n;
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        //out.write("Welcome user "+request.getParameter("nam"));
        n=request.getParameter("nam");
        response.sendRedirect("http://www."+n);
    }
}

```

195. Write a program to give name of author in servlet config parameter and display.

Web.xml

```

<servlet>
    <servlet-name>servlet1</servlet-name>
    <servlet-class>servlet1</servlet-class>
    <init-param>
        <param-name>author</param-name>
        <param-value>Swati saxena</param-value>
    </init-param>
</servlet>

```

Servlet1

```

import java.io.IOException;
import java.io.PrintWriter;

```

```
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
public class servlet1 extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        ServletConfig conf=getServletConfig();
        out.println(conf.getInitParameter("author"));
    }
}
```

1. A publisher publishes books for many authors. Write a servlet to write publisher name in context parameter and display.

Web.xml

```
<web-app ...>

    <context-param>
        <param-name>publisher</param-name>
        <param-value>bpb </param-value>
    </context-param>

    <servlet>
        <servlet-name>servlet1</servlet-name>
        <servlet-class>servlet1</servlet-class>
    </servlet>
```

```
</web-app>
```

servlet1

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class servlet1 extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse
    response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        ServletContext sc = getServletContext();
        out.println("publisher of book is :"+sc.getInitParameter("publisher"));
    }
}
```

Session Tracking Techniques

There are four techniques used in the Session tracking:

1. **HttpSession**
2. **Cookies**
3. **Hidden Form Field**
4. **URL Rewriting**

196. Write a servlet to create session when user login and destroy session when logout.

Form1.html

```
<html>
  <head>
    <title>TODO supply a title</title>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width">
</head>
<body>
  <form action=".servlet1" method="post">
    Enter name<input type="text" name="nam"><br>
    Enter password<input type="password" name="pwd">
    <input type="submit" value="Login">
  </form>
</body>
</html>
Logout.html
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
  </head>
  <body>
    <form action=".logout" method="post">
      Are you sure to Logout <input type="submit" value="yes">
    </form>
  </body>
</html>
```

Servlet1

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```



```

public class servlet1 extends HttpServlet {

    String n,p;
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        RequestDispatcher rd;
        n=request.getParameter("nam");
        p=request.getParameter("pwd");
        if(n.equals(p))
        {
            HttpSession ses=request.getSession(true);
            ses.setAttribute("user", n);
            rd=request.getRequestDispatcher("servlet2");
            rd.forward(request, response);;
        }
        else
        {
            rd=request.getRequestDispatcher("form1.html");
            rd.forward(request, response);;
        }
    }
}

```

Servlet2

```

public class servlet2 extends HttpServlet {
    protected void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        HttpSession ses=req.getSession(false);
        res.setContentType("text/html");
        PrintWriter pw=res.getWriter();
    }
}

```

```
        pw.write("welcome user " +ses.getAttribute("user") );
        pw.write("<a href=logout.html>logout</a>");
    }
}
```

Logout.java

```
public class logout extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req,HttpServletResponse
    res)throws ServletException, IOException
    {
        HttpSession ses=req.getSession(false);
        ses.invalidate();
        res.sendRedirect("form1.html");
    }
}
```

197. Write a servlet to create a cookie file to save user name in cookie file and display?

MyServlet

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MyServlet extends HttpServlet {

    protectedvoiddoPost(HttpServletRequestrequest,HttpServletResponse
    response)
        throws ServletException, IOException {
        response.setContentType("text/html ");
        String name = request.getParameter("user");
        String pass = request.getParameter("pass");

        if(pass.equals("1234"))
        {
```

```

        Cookie ck = new Cookie("username",name);
        response.addCookie(ck);
        response.sendRedirect("First");
    }
}

```

First

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

```

```

public class First extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse
    response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Cookie[] cks = request.getCookies();
        out.println("Welcome "+cks[0].getValue());
    }
}

```

198. Write a servlet to save user name for multiple servlet in hidden field and display.

Form1.html

```

<form method="post" action=".\\servlet1">
    Name:<input type="text" name="user" /><br/>
    Password:<input type="text" name="pass" /><br/>
    <input type="submit" value="submit">
</form>
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

```

Servlet1

```
public class Servlet1 extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse
    response)
        throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out = response.getWriter();

        //getting value submitted in form from HTML file
        String user = request.getParameter("user");

        //creating a new hidden form field
        out.println("<form action='Second'>");
        out.println("<input type='hidden' name='user' value='"+user+"'>");
        out.println("<input type='submit' value='submit' >");
        out.println("</form>");
    }
}
```

Second.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Second extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse
    response)
        throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out = response.getWriter();

        //getting parameter from the hidden field
        String user = request.getParameter("user");
```

```

        out.println("Welcome "+user);
    }
}

```

199. Write a servlet to pass company name of selected mobile in URL.

Form1.html

```

<form action=".\\servlet1" method="get">
    <a href="servlet1?comp=nokia"></a><br>
    <a href="servlet1?comp=samsung"></a>
</form>

```

Servlet1

```

public class servlet1 extends HttpServlet {

    String n,p;
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
    response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        RequestDispatcher rd;
        p=request.getParameter("comp");
        out.print(p);
    }
}

```

200. Write a servlet to count hits on a button.

Form1.html

```

<html>
<head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
</head>
<body>

```

```
<form action=".\\servlet1" method="post" >
    <input type="submit" value="Login">
</form>
</body>
</html>
```

Servlet1

```
public class servlet1 extends HttpServlet {
```

```
    String n,p;
    int hitcount;
    public void init()
    {
        hitcount=0;
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        hitcount++;
        RequestDispatcher rd=request.getRequestDispatcher("form1.html");
        out.println("Hits:"+hitcount);
        rd.include(request, response);
    }
}
```

201. Write a program to upload your resume .

```
<html>
<body>
<form action=".\\servlet1" method="post" enctype="multipart/form-data">
Select File:<input type="file" name="fname"/><br/>
<input type="submit" value="upload"/>
</form>
```

```
</body>
</html>
```

Servlet1

```
import java.io.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;
import com.oreilly.servlet.MultipartRequest;

public class servlet1 extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse
    response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        MultipartRequest m=new MultipartRequest(request,"d:/new");
        out.print("successfully uploaded");
    }
}
```

202. Write a program to write an image using ServletOutputStream.

```
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width">
    </head>
    <body>
        <form action=".\\servlet1" method="get" >
            Enter name<input type="text" name="nam"><br>
            Enter password<input type="password" name="pwd">
            <input type="submit" value="Login">
```

```
        </form>
    </body>
</html>
```

Servlet1

```
public class servlet1 extends HttpServlet {

    @Override

    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("image/jpeg");
        ServletOutputStream out;
        out = response.getOutputStream();
        FileInputStream fin = new FileInputStream("f:\\image\\1.jpg");
        int ch =0; ;
        while((ch=fin.read())!=-1)
        {
            out.write(ch);
        }

        fin.close();

        out.close();
    }
}
```

203. Write a program to enter student roll number, name, and marks in the form, validate it using servlet and insert record in database?

```
<html>
    <title>Validation Project</title>
    <head><center><h2>Student form validation</h2></center></head>
    <body>
```



```

<center>
  <form " action="./Validation" method="post">
    <table border="1">
      <tr>
        <th>Enter student number : </th>
        <td><input type="text" name="validpro_sno" value=""></td>
      </tr>
      <tr>
        <th>Enter student name : </th>
        <td><input type="text" name="validpro_sname" value=""></td>
      </tr>
      <tr>
        <th>Enter student marks : </th>
        <td><input type="text" name="validpro_smarks" value=""></td>
      </tr>
      <tr>
        <td colspan="2">
          <td><input type="submit" value="Insert"></td>
          <td><input type="reset" value="Clear"></td>
        </td>
      </tr>
    </table>
  </form>
</center>
</body>
</html>

```

Validation.java

```

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

```

```
import java.sql.*;
import java.util.*;

public class Validation extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        ArrayList al = new ArrayList();
        res.setContentType("text/html");
        PrintWriter pw = res.getWriter();
        String sno1 = req.getParameter("validpro_sno");
        String sname = req.getParameter("validpro_sname");
        String smarks1 = req.getParameter("validpro_smarks");
        int sno = 0;
        float smarks = 0;
        if ((sno1 == null) || (sno1.equals("")))) {
            al.add("PROVIDE STUDENT NUMBER...");
        } else {
            try {
                sno = Integer.parseInt("sno1");
            } catch (NumberFormatException nfe) {
                al.add("PROVIDE int DATA IN STUDENT NUMBER...");
            }
        }
        if ((sname == null) || (sname.equals("")))) {
            al.add("PROVIDE STUDENT NAME...");
        }
        if ((smarks1 == null) || (smarks1.equals("")))) {
            al.add("PROVIDE STUDENT MARKS...");
        } else {
            try {
                smarks = Float.parseFloat("smarks1");
            } catch (NumberFormatException nfe) {
                al.add("PROVIDE float DATA IN STUDENT MARKS...");
            }
        }
    }
}
```

```

    }
    if (al.size() != 0) {
        pw.println(al);
    } else {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:Hanuman", "scott", "tiger");
            PreparedStatement ps = con.prepareStatement("insert into Student values (?, ?, ?)");
            ps.setInt(1, sno);
            ps.setString(2, sname);
            ps.setFloat(3, smarks);
            int i = ps.executeUpdate();
            if (i > 0) {
                pw.println("RECORD INSERTED...");
            } else {
                pw.println("RECORD NOT INSERTED...");
            }
            con.close();
        } catch (Exception e) {
            res.sendError(503, "PROBLEM IN DATABASE...");
        }
    }
}

public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
    doGet(req, res);
}
}

```

204. Write a program to explain the use of servlet filter.

Web.xml
<filter>

```
<filter-name>LogFilter</filter-name>
<filter-class>LogFilter</filter-class>
<init-param>
  <param-name>test-param</param-name>
  <param-value>Initialization Paramter</param-value>
</init-param>
</filter>
```

```
<filter-mapping>
  <filter-name>LogFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

LogFilter.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

// Implements Filter class
public class LogFilter implements Filter {
    public void init(FilterConfig config) throws ServletException {

        // Get init parameter
        String testParam = config.getInitParameter("test-param");

        //Print the init parameter
        System.out.println("Test Param: " + testParam);
    }

    public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain chain) throws java.io.IOException, ServletException {

        // Get the IP address of client machine.
        String ipAddress = request.getRemoteAddr();
```

```

// Log the IP address and current timestamp.
System.out.println("IP " + ipAddress + ", Time " + new Date().
toString());

// Pass request back down the filter chain
chain.doFilter(request,response);
}

public void destroy( ) {
    /* Called before the Filter instance is removed from service by the
    web container*/
}
}

```

205. Write a program to display current time and auto refresh it.

```

public class servlet1 extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse
    response)
    throws ServletException, IOException {

        // Set refresh, autoloading time as 1 seconds
        response.setIntHeader("Refresh", 1);

        response.setContentType("text/html");

        // Get current time
        Calendar calendar = new GregorianCalendar();
        String am_pm;
        int hour = calendar.get(Calendar.HOUR);
        int minute = calendar.get(Calendar.MINUTE);
        int second = calendar.get(Calendar.SECOND);

        if(calendar.get(Calendar.AM_PM) == 0)
            am_pm = "AM";

```

```
else
```

```
    am_pm = "PM";
```

```
String CT = hour+":"+ minute +":"+ second + "+" + am_pm;
```

```
PrintWriter out = response.getWriter();
```

```
String title = "Auto Page Refresh using Servlet";
```

```
String docType =
```

```
    "<!doctype html public \"-//w3c//dtd html 4.0 \" + \"transitional//  
    en\">\n\";
```

```
out.println(docType +
```

```
    "<html>\n" +
```

```
    "<head><title>" + title + "</title></head>\n" +
```

```
    "<body bgcolor = \"#f0f0f0\">\n" +
```

```
    "<h1 align = \"center\">" + title + "</h1>\n" +
```

```
    "<p>Current Time is: " + CT + "</p>\n"
```

```
);
```

```
}
```

```
// Method to handle POST method request.
```

```
public void doPost(HttpServletRequest request, HttpServletResponse  
response)
```

```
    throws ServletException, IOException {
```

```
    doGet(request, response);
```

```
}
```

```
}
```

JSP

JSP technology is used to create web application just like Servlet technology.

This can also be used to access JavaBeans objects.

The jsp pages are easier to maintain than servlet because we can separate designing and development.

It provides some additional features such as Expression Language, Custom Tag etc.

We can share information across pages using request and response objects.

JSP can be used for separation of the view layer with the business logic in the web application.

If JSP page is modified, we don't need to recompile and redeploy the project.

The servlet code needs to be updated and recompiled if we have to change the look and feel of the application.

JSP scripting Element:

1. scriptlet tag `<% ..... %>`
2. expression tag `<%= ..... %>`
3. declaration tag `<%!.....%>`

206. Write a JSP page to print square of a given number.

```
<html>
<body>
<%!
int square(int n){
return n*n;
}
%>
<%= "square of 5 is :"+square(5) %>
</body>
```

```
</html>
```

Write a JSP page to display current date and time?

```
<html>
<body>
<% out.print("Today is:"+java.util.Calendar.getInstance().getTime()); %>
</body>
</html>
```

207. Write a JSP page to print welcome message , take name as input in HTML form?

```
<html>
<body>
<form action="welcome.jsp">
<input type="text" name="uname">
<input type="submit" value="message for me"><br/>
</form>
</body>
</html>
```

Welcome.jsp

```
<html>
<body>
<%
String name=request.getParameter("uname");
out.print("welcome "+name);
%>
</form>
</body>
</html>
```

208. Write a JSP page to display all input fields in HTML form.

```
<html>
<head>
<metahttp-equiv="Content-Type"content="text/html;charset=UTF-8">
```



```

<title>JSP Page</title>
</head>
<body>
  <form name="frm" method="get" action="user.jsp">
    Enter name<input type="text" name="un"><br>
    Enter password<input type="password" name="pwd"><br>
    Select gender <input type="radio" name="gen" value="male">male
    <input type="radio" name="gen" value="female">female<br>
    Language known:<input type="checkbox" value="english"
    name="en">Eng
    <input type="checkbox" value="hindi" name="hi">hin
    <input type="checkbox" value="french" name="fr">Fre

    <input type="submit" value="Register">

  </body>
</html>

```

User.jsp

```

<%@page      import="java.util.*;"      contentType="text/html"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta      http-equiv="Content-Type"      content="text/html;
    charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <%
Enumeration e=request.getParameterNames();
while(e.hasMoreElements())
{
  String n=e.nextElement().toString();
  out.println(request.getParameter(n));

```

```
}  
%>  
</body>  
</html>
```

209. Write a JSP page to login user and create session, on click, logout session should destroy and home page should display.

Index.jsp

```
<html>  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
    <title>JSP Page</title>  
  </head>  
  <body>  
    <form name="frm" method="get" action="user.jsp">  
      Enter name<input type="text" name="un"><br>  
      Enter password<input type="password" name="pwd"><br>  
      <input type="submit" value="Login">  
  
    </body>  
</html>
```

User.jsp

```
<html>  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
    <title>JSP Page</title>  
  </head>  
  <body>  
    <%  
      session.setAttribute("uname", request.getParameter("un"));  
  
      out.print("<a href='next.jsp'>Logout</a>");  
    %>
```

```

    </body>
</html>

```

Next.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <%
            session.invalidate();
            response.sendRedirect("index.jsp");
        %>
    </body>
</html>

```

210. Write a JSP page to send email.

```

<%@ page import = "java.io.*,java.util.*,javax.mail.*"%>
<%@ page import = "javax.mail.internet.*,javax.activation.*"%>
<%@ page import = "javax.servlet.http.*,javax.servlet.*"%>

<%
    String status;

    // Recipient's email ID needs to be mentioned.
    String to = "swaticomputers01@gmail.com";

    // Sender's email ID needs to be mentioned
    String from = "sc01@gmail.com";

    // Assuming you are sending email from localhost

```

```
String host = "localhost";

// Get system properties object
Properties properties = System.getProperties();

// Setup mail server
properties.setProperty("mail.smtp.host", host);

// Get the default Session object.
Session mailSession = Session.getDefaultInstance(properties);

try {
    // Create a default MimeMessage object.
    MimeMessage message = new MimeMessage(mailSession);

    // Set From: header field of the header.
    message.setFrom(new InternetAddress(from));

    // Set To: header field of the header.
    message.addRecipient(Message.RecipientType.TO,
        new InternetAddress(to));
    // Set Subject: header field
    message.setSubject("This is the Subject Line!");

    // Now set the actual message
    message.setText("This is actual message");

    // Send message
    Transport.send(message);
    status = "Sent message successfully....";
} catch (MessagingException mex) {
    mex.printStackTrace();
    status = "Error: unable to send message....";
}
%>
```

```

<html>
  <head>
    <title>Send Email using JSP</title>
  </head>

  <body>
    <center>
      <h1>Send Email using JSP</h1>
    </center>

    <p align = "center">
      <%
        out.println("status: " + status + "\n");
      %>
    </p>
  </body>
</html>

```

211. Write different programs to upload and display photos.

Create HTML form to upload photo.

Create a HTML form to take data regarding a new photo. It contains two text fields and one file field to upload photo to server.

```

<html>
  <head>
    <title>Add Photo</title>
  </head>
  <body>
    <h2>Add Photo</h2>
    <formid="form1"enctype="multipart/form-data"action="addphoto"
    method="post">
      <table>
        <tr>
          <td>Enter Photo Id :</td>
          <td><input type="text" name="id"/></td>
        </tr>

```

```
<tr>
    <td>Enter Title For Photo :</td>
    <td><input type="text" name="title"/></td>
</tr>
<tr>
    <td>Select Photo </td>
    <td><input type="file" name="photo" />
</tr>
</table>
<p/>
<input type="submit" value="Add Photo"/>
</form>

<p/>
<a href="listphotos">List Photos </a>
</body>
</html>
```

AddPhotoServlet

When user clicks on **Submit** button, the above form calls AddPhotoServlet, which inserts a row into PHOTOS table. The code for AddPhotoServlet.java is given below. Create this servlet and assign addphoto as the url pattern for it.

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.commons.fileupload.FileItem;
```

```
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;

public class AddPhotoServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            // Apache Commons-Fileupload library classes
            DiskFileItemFactory factory = new DiskFileItemFactory();
            ServletFileUpload sfu = new ServletFileUpload(factory);

            if (! ServletFileUpload.isMultipartContent(request)) {
                System.out.println("sorry. No file uploaded");
                return;
            }

            // parse request
            List items = sfu.parseRequest(request);
            FileItem id = (FileItem) items.get(0);
            String photoid = id.getString();

            FileItem title = (FileItem) items.get(1);
            String phototitle = title.getString();

            // get uploaded file
            FileItem file = (FileItem) items.get(2);

            // Connect to Oracle
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:
thin:@localhost:1521:xe", "hr", "hr");
            con.setAutoCommit(false);
```

```
PreparedStatement ps = con.prepareStatement("insert into photos
values(?,?,?)");
    ps.setString(1, photoid);
    ps.setString(2, phototitle);
    // size must be converted to int otherwise it results in error
    ps.setBinaryStream(3, file.getInputStream(), (int) file.getSize());
    ps.executeUpdate();
    con.commit();
    con.close();
    out.println("Photo Added Successfully. <p> <a href='listphotos'>List
    Photos </a>");
}
catch(Exception ex) {
    out.println( "Error --> " + ex.getMessage());
}
}
```

ListPhotosServlet

Create another servlet - ListPhotosServlet.java - to display the list of photos from PHOTOS table. It uses DisplayImageServlet to display each photo.

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ListPhotosServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
```



```

    HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection
("jdbc:oracle:thin:@localhost:1521:xe", "hr", "hr");
            PreparedStatement ps =
con.prepareStatement("select * from photos");
            ResultSet rs = ps.executeQuery();
            out.println("<h1>Photos</h1>");
            while ( rs.next()) {
                out.println("<h4>" + rs.getString("title") + "</h4>");
                out.println("<img width='600' height=
'600' src=displayphoto?id
=" + rs.getString("id") + "></img> <p/>");
            }

            con.close();
        }
        catch(Exception ex) {

        }
        finally {
            out.close();
        }
    }

    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }
}

```

```
@Override
protected void doPost(HttpServletRequest request,
HttpServletRequest response)
throws ServletException, IOException {
    processRequest(request, response);
}
}
```

DisplayPhotoServlet

DisplayPhotoServlet.java sends a a single photo that is taken from PHOTO column of PHOTOS table to browser as outputstream. This servlet takes id as parameter (querystring) and sends photo to IMG tag.

```
DisplayPhotoServlet.java
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.Blob;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
public class DisplayPhotoServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
HttpServletRequest response)
        throws ServletException, IOException {

        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection
            ("jdbc:oracle:thin:@localhost:1521:xe", "hr", "hr");
```

```

        PreparedStatement ps = con.prepareStatement
("select photo from photos where id = ?");
        String id = request.getParameter("id");
        ps.setString(1,id);
        ResultSet rs = ps.executeQuery();
        rs.next();
        Blob b = rs.getBlob("photo");
        response.setContentType("image/jpeg");
        response.setContentLength( (int) b.length());
        InputStream is = b.getBinaryStream();
        OutputStream os = response.getOutputStream();
        byte buf[] = new byte[(int) b.length()];
        is.read(buf);
        os.write(buf);
        os.close();
    }
    catch(Exception ex) {
        System.out.println(ex.getMessage());
    }
}

```

```

@Override
protected void doGet(HttpServletRequest request,
HttpServletRequest response)
throws ServletException, IOException {
    processRequest(request, response);
}
@Override
protected void doPost(HttpServletRequest request,
HttpServletRequest response)
throws ServletException, IOException {
    processRequest(request, response);
}
}

```

Here is web.xml file that contains entries related to Servlets.

```
<servlet>
  <servlet-name>AddPhoto</servlet-name>
  <servlet-class>AddPhotoServlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>ListPhotos</servlet-name>
  <servlet-class>ListPhotosServlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>DisplayPhotoServlet</servlet-name>
  <servlet-class>DisplayPhotoServlet</servlet-class>
</servlet>
<servlet-mapping>

<servlet-name>AddPhoto</servlet-name>
  <url-pattern>/addphoto</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>ListPhotos</servlet-name>
  <url-pattern>/listphotos</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>DisplayPhotoServlet</servlet-name>
  <url-pattern>/displayphoto</url-pattern>
</servlet-mapping>
```

After all servlets are created, build the project and deploy it. Then run `addphoto.html` and enter details of some photos. Then you can click on List Photos link to see the list of photos placed in the database.

212. Write two JApplet and display inter applet communication.

First.java

```
import java.applet.Applet;
import javax.swing.JApplet;
import java.applet.AppletContext;
```

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class first extends JApplet implements ActionListener
{
    JButton b,b1,b2;
    Color c;
    public void init()
    {
        setLayout(new FlowLayout());
        b=new JButton("Red");
        b1=new JButton("Green");
        b2=new JButton("Blue");
        add(b);add(b1);add(b2);
        b.addActionListener(this);
        b1.addActionListener(this);
        b2.addActionListener(this);
    }
    public Color getCol()
    {
        return c;
    }
    public void set(Color c)
    {
        this.c=c;
    }
    public void actionPerformed(ActionEvent ae)
    {
        if(ae.getSource()==b)
            c=new Color(255,0,0);
        else if(ae.getSource()==b1)
            c=new Color(0,255,0);
        else if(ae.getSource()==b2)
            c=new Color(0,0,255);
    }
}
```

```
second f =  
    (second)getAppletContext().getApplet("Second");  
if(f!=null)f.repaint();
```

```
}  
}
```

Second.java

```
import java.applet.Applet;  
import javax.swing.JApplet;  
import javax.swing.*;  
import java.awt.event.*;  
import java.awt.*;  
public class second extends JApplet  
{  
    Color c;  
    public void init()  
    {  
        c=new Color(0,0,0);  
    }  
    public void paint(Graphics g)  
    {  
        //g.setColor(Color.pink);  
        first f =  
            (first)getAppletContext().getApplet("First");  
        if(f!=null)  
        {  
            c=f.getCol();  
        }  
        g.setColor(c);  
  
        g.fillOval(50,50,100,100);  
    }  
}
```

EXTRA EFFORTS:

RMI, Border in Swing component, LookAndFeel etc.

213. What is RMI?

Remote Method Invocation (RMI)

The RMI is an API that provides a mechanism to create distributed application in java.

The RMI allows an object to invoke methods on an object running in another JVM.

The RMI provides remote communication between the applications using two objects stub and skeleton.

The RMI is distributed application because :

1. The application need to locate the remote method.
2. It need to provide the communication with the remote objects.
3. The application need to load the class definitions for the objects.

214. What is pluggable Look and feel?

Java Swing api provides **pluggable look and feel (PL&F)** capability that allows swing GUI widgets to change appearance based on the programmers customized look and feel setting.

Each Java runtime has a UIManager object that determines the look and feel of the Swing components.

215. Write a program to display the movement of mouse wheel.

```
import java.awt.BorderLayout;
import java.awt.Dimension;

import java.awt.event.MouseWheelEvent;
```

```
import java.awt.event.MouseWheelListener;

import javax.swing.BorderFactory;

import javax.swing.JComponent;
import javax.swing.JFrame;

import javax.swing.JPanel;
import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class aa extends JPanel implements MouseWheelListener {
    JTextArea textArea = new JTextArea();

    JScrollPane scrollPane = new JScrollPane(textArea);

    final static String newline = "\n";

    public aa() {
        super(new BorderLayout());
        textArea.setEditable(false);

        scrollPane.setVerticalScrollBarPolicy(JScrollPane.
VERTICAL_SCROLLBAR_ALWAYS);
        scrollPane.setPreferredSize(new Dimension(400, 250));
        add(scrollPane, BorderLayout.CENTER);
        textArea.append("This text area displays information
+ "about its mouse wheel events."
+ newline);

        textArea.addMouseWheelListener(this);

        setPreferredSize(new Dimension(450, 350));
```



```

    setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
}

public void mouseWheelMoved(MouseWheelEvent e) {
    String message;
    int notches = e.getWheelRotation();
    if (notches < 0) {
        message = "Mouse wheel moved UP " + -notches + " notch(es)" +
            newline;
    } else {
        message = "Mouse wheel moved DOWN " + notches + " notch(es)" +
            newline;
    }
    if (e.getScrollType() == MouseWheelEvent.WHEEL_UNIT_SCROLL) {
        message += " Scroll type: WHEEL_UNIT_SCROLL" + newline;
        message += " Scroll amount: " + e.getScrollAmount() + " unit
            increments per notch"
            + newline;
        message += " Units to scroll: " + e.getUnitsToScroll() + " unit
            increments" + newline;
        message += " Vertical unit increment: "
            + scrollPane.getVerticalScrollBar().getUnitIncrement(1) + "pixels"
            + newline;
    } else { // scroll type == MouseWheelEvent.WHEEL_BLOCK_SCROLL
        message += " Scroll type: WHEEL_BLOCK_SCROLL" + newline;
        message += " Vertical block increment: "
            + scrollPane.getVerticalScrollBar().getBlockIncrement(1) + "
            pixels" + newline;
    }
    saySomething(message, e);
}

void saySomething(String eventDescription, MouseWheelEvent e) {

```

```
        textArea.append(e.getComponent().getClass().getName() + ": " +
        eventDescription);
        textArea.setCaretPosition(textArea.getDocument().getLength());
    }
    public static void main(String[] args) {
        JFrame frame = new JFrame("MouseWheelEventDemo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JComponent newContentPane = new aa();
        newContentPane.setOpaque(true);
        frame.setContentPane(newContentPane);

        frame.pack();
        frame.setVisible(true);
    }
}
```

216. RMI: Create a remote interface for additon of two numbers. Create another class extends UnicastRemoteObject to implement remote interface, create client class to call method?

interface

```
import java.rmi.*;
public interface Adder extends Remote{
    public int add(int x,int y)throws RemoteException;
}
```

Implements remote interface

```
import java.rmi.*;
import java.rmi.server.*;
public class AdderRemote extends UnicastRemoteObject implements
Adder{
    AdderRemote()throws RemoteException{
        super();
    }
    public int add(int x,int y){return x+y;}
}
```

Bind method with name

```
import java.rmi.*;
import java.rmi.registry.*;
public class MyServer
{
    public static void main(String args[]){
        try{
            Adder stub=new AdderRemote();
            Naming.rebind("rmi://localhost:5000/sonoo",stub);
        }
        catch(Exception e)
        {System.out.println(e);}
    }
}
```

Client to call remote method

```
import java.rmi.*;
public class MyClient{
    public static void main(String args[]){
        try{
            Adder stub=(Adder)Naming.lookup("rmi://localhost:5000/sonoo");
            System.out.println(stub.add(34,4));
        }
        catch(Exception e){}
    }
}
```

217. Write a program to change the border on clicking on the button?

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.util.*;
class fd1 extends JFrame implements ActionListener {
    Border bor;
    FileDialog f;
```

```

JButton b,b1,b2,b3,b4;
JPanel p;

public fd1(String tit)
{
    super(tit);
    double n=(Math.random()*100);
    System.out.println(n);
    p=new JPanel();
    b=new JButton("click");
    b1=new JButton("click");
    b2=new JButton("click");
    b3=new JButton("click");
    b4=new JButton("click");
    p.add(b);
    getContentPane().add(p);
    getContentPane().add(b1,"North");
    getContentPane().add(b2,"South");
    getContentPane().add(b3,"East");
    getContentPane().add(b4,"West");
    b.addActionListener(this);
    b1.addActionListener(this);
    b2.addActionListener(this);
    b3.addActionListener(this);
    b4.addActionListener(this);

    setSize(200,200);
    setVisible(true);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
}

public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==b)
    {
        f=new FileDialog(this, "open file", FileDialog.LOAD);
    }
}
```

```

        f.setSize(200,200);
        f.setVisible(true);
    }
    if(ae.getSource()==b1)
    {
        bor=BorderFactory.createLineBorder(Color.yellow, 5);
        p.setBorder(bor);
    }
    if(ae.getSource()==b2)
    {
        bor=BorderFactory.createBevelBorder(BevelBorder.RAISED,
        Color.green, Color.yellow, Color.blue, Color.pink);
        p.setBorder(bor);
    }
    if(ae.getSource()==b3)
    {
        bor=BorderFactory.createTitledBorder("hello");
        b.setBorder(bor);
    }
    if(ae.getSource()==b4)
    {
        bor=BorderFactory.createMatteBorder(10, 10, 10, 10, new
        ImageIcon("1.jpg"));
        b.setBorder(bor);
    }
}

class fd{
    public static void main(String [] arg)
    {
        new fd1("hi");
    }
}

```

218. Write a program to change look and feel of components on button click?

```
import java.util.*;
import com.sun.java.swing.plaf.motif.MotifLookAndFeel;
import com.sun.java.swing.plaf.windows.WindowsLookAndFeel;
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;
import javax.swing.plaf.metal.MetalLookAndFeel;
import javax.swing.plaf.nimbus.NimbusLookAndFeel;
class look extends JFrame implements ActionListener
{
    JButton b,b1,b2,b3,b4;
    MetalLookAndFeel mf;
    WindowsLookAndFeel wf;
    MotifLookAndFeel mof;
    NimbusLookAndFeel nf;
    JPanel p;
    public look(String tit)
    {
        super(tit);
        b=new JButton("click");
        p=new JPanel();
        b1=new JButton("click");
        b2=new JButton("click");
        b3=new JButton("click");
        b4=new JButton("click");
        p.add(b);
        //b.setForeground(Color.red);
        getContentPane().add(p);
        getContentPane().add(b1,"North");
        getContentPane().add(b2,"South");
```

```

    getContentPane().add(b3,"East");
    getContentPane().add(b4,"West");
    b.addActionListener(this);
    b1.addActionListener(this);
    b2.addActionListener(this);
    b3.addActionListener(this);
    b4.addActionListener(this);

    setSize(200,200);
    setVisible(true);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==b1)
    {
        mf=new MetalLookAndFeel();
        try {

            UIManager.setLookAndFeel(mf);
            SwingUtilities.updateComponentTreeUI(this);
        } catch (UnsupportedLookAndFeelException ex) {
            Logger.getLogger(look.class.getName()).log(Level.SEVERE,null,
            ex);
        }
    }

    if(ae.getSource()==b2)
    {
        wf=new WindowsLookAndFeel();
        try {

            UIManager.setLookAndFeel(wf);
            //SwingUtilities.updateComponentTreeUI(this);
            for(Component c:this.getContentPane().getComponents())

```

```
        {
        if(c instanceof JButton)
        {
            System.out.println("compo"+c);
            c.setBackground(Color.yellow);
        }
        if(c instanceof JPanel)
        {
            c.setBackground(Color.pink);
        }
        }
        //((JButton)ae.getSource()).setBackground(Color.red);
    } catch (UnsupportedLookAndFeelException ex) {
        Logger.getLogger(look.class.getName()).log(Level.SEVERE, null, ex);
    }

}

if(ae.getSource()==b3)
{
    mof=new MotifLookAndFeel();
    try {

        UIManager.setLookAndFeel(mof);
        SwingUtilities.updateComponentTreeUI(this);
    } catch (UnsupportedLookAndFeelException ex) {
        Logger.getLogger(look.class.getName()).log(Level.SEVERE, null, ex);
    }

}

if(ae.getSource()==b4)
{
    nf=new NimbusLookAndFeel();
    try {
```



```

        UIManager.setLookAndFeel(nf);
        SwingUtilities.updateComponentTreeUI(this);
    } catch (UnsupportedLookAndFeelException ex) {
        Logger.getLogger(look.class.getName()).log(Level.SEVERE, null, ex);
    }

}
}
}
public class looks {
    public static void main(String [] arg)
    {
        new look("hi");
    }
}

```

219. Write a JFrame for School project with principal, teacher, and student tabs, in tabs create different buttons to give functionality.

```

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*;
import com.sun.java.swing.plaf.motif.MotifLookAndFeel;
import com.sun.java.swing.plaf.windows.WindowsLookAndFeel;
import java.util.logging.Level;
import java.util.logging.Logger;
class school extends JFrame implements ActionListener,MouseListener

{
    JTabbedPane tp;
    JPanel p,t,s;
    JButton stu,teac,per,notes,stuv,ok;
    JTextField a,b,c,d,e,f,g,h;
    JLabel l1,l2,l3,l4,l5,l6,l7,l8;
    JDialog d1,d2;

```

```
MotifLookAndFeel mof;  
WindowsLookAndFeel wlf;
```

```
public school(String tit)  
{  
    super(tit);  
    stu=new JButton("Student");  
    teac=new JButton("Teacher");  
    per=new JButton("personal info");  
    notes=new JButton("Notes");  
    stuv=new JButton("student window");  
    tp=new JTabbedPane();  
    p=new JPanel();  
    ok=new JButton("Ok");  
    t=new JPanel();  
    s=new JPanel();  
    l1=new JLabel("Name");  
    l2=new JLabel("Class");  
    l3=new JLabel("Roll No.");  
    l4=new JLabel("Attendance");  
    l5=new JLabel("Grades");  
    a=new JTextField(10);  
    b=new JTextField(10);  
    c=new JTextField(10);  
    d=new JTextField(10);  
    e=new JTextField(10);  
    l6=new JLabel("Name");  
    l7=new JLabel("Attendance");  
    l8=new JLabel("Complains");  
    f=new JTextField(10);  
    g=new JTextField(10);  
    h=new JTextField(10);  
  
    tp.add(p,"princpal");  
    tp.add(t,"teacher");
```

```
tp.add(s,"student");
p.setLayout(null);
p.add(stu);
p.add(teac);

stu.setBounds(10,100,90,30);
teac.setBounds(100,100,90,30);

t.setLayout(null);
t.add(per);
t.add(notes);
per.setBounds(10,100,90,30);
notes.setBounds(100,100,90,30);

s.setLayout(null);
s.add(stuv);
stuv.setBounds(10,100,140,30);
stu.addActionListener(this);
teac.addActionListener(this);
per.addActionListener(this);
stuv.addActionListener(this);
ok.addActionListener(this);
per.addActionListener(this);
t.addMouseListener(this);
getContentPane().add(tp);
setSize(500,500);
setVisible(true);
setDefaultCloseOperation(EXIT_ON_CLOSE);

}

public void mouseEntered(MouseEvent me){
    try {
        mof=new MotifLookAndFeel();
        UIManager.setLookAndFeel(mof);
        SwingUtilities.updateComponentTreeUI(this);
```

```
    } catch (UnsupportedLookAndFeelException ex) {
        Logger.getLogger(school.class.getName()).log(Level.SEVERE,
            null, ex);
    }
}

public void mouseClicked(MouseEvent me){
    try {
        mof=new MotifLookAndFeel();
        UIManager.setLookAndFeel(mof);
        SwingUtilities.updateComponentTreeUI(this);
    } catch (UnsupportedLookAndFeelException ex) {
        Logger.getLogger(school.class.getName()).log(Level.SEVERE,
            null, ex);
    }
}

public void mouseExited(MouseEvent me){}
public void mousePressed(MouseEvent me){}
public void mouseReleased(MouseEvent me){}
public void actionPerformed(ActionEvent ae)
{
    if(ae.getSource()==per)
    {
        mof=new MotifLookAndFeel();
        try {

            UIManager.setLookAndFeel(mof);
            SwingUtilities.updateComponentTreeUI(this);
        } catch (UnsupportedLookAndFeelException ex) {
            Logger.getLogger(school.class.getName()).log(Level.SEVERE,
                null, ex);
        }
    }
}

if(ae.getSource()==stu)
{
```

```
wlf= new WindowsLookAndFeel();
try {

    UIManager.setLookAndFeel(wlf);
    SwingUtilities.updateComponentTreeUI(this);
} catch (UnsupportedLookAndFeelException ex) {
    Logger.getLogger(school.class.getName()).log(Level.SEVERE,
    null, ex);
}

    d1=new JDialog(this,"hi",true);
    d1.setLayout(null);
    d1.add(ok);
ok.setBounds(50,200,50,20);

    d1.add(l1);
    l1.setBounds(20,20,70,20);
    d1.add(l2);
    l2.setBounds(20,50,100,20);
    d1.add(l3);
    l3.setBounds(20,80,100,20);
    d1.add(l4);
    l4.setBounds(20,110,100,20);
    d1.add(l5);
    l5.setBounds(20,140,100,20);
    d1.add(a);
    a.setBounds(100,20,70,20);
    d1.add(b);
    b.setBounds(100,50,100,20);
    d1.add(c);
    c.setBounds(100,80,100,20);
    d1.add(d);
    d.setBounds(100,110,100,20);
    d1.add(e);
    e.setBounds(100,140,100,20);
```

```
        d1.setSize(400,400);
        d1.setVisible(true);

    }

    if(ae.getSource()==ok)
    {
        d1.setVisible(false);

    }
    if(ae.getSource()==teac)
    {

        d2=new JDialog(this,"hi",true);
        d2.setLayout(null);
        d2.add(f);
        f.setBounds(100,20,100,20);
        d2.add(g);
        g.setBounds(100,50,100,20);
        d2.add(h);
        h.setBounds(100,80,100,20);
        d2.add(l6);
        l6.setBounds(20,20,70,20);
        d2.add(l7);
        l7.setBounds(20,50,100,20);
        d2.add(l8);
        l8.setBounds(20,80,100,20);

        d2.setSize(400,400);
        d2.setVisible(true);

    }

    if(ae.getSource()==per)
    {

        String str=f.getText();
```

```

String str1=g.getText();

String str2=h.getText();
JOptionPane.showMessageDialog(this, “ “+ str+” “+str1+”%”+” “+str2 );

}
if(ae.getSource()==stuv)
{

String str=a.getText();
String str1=b.getText();
String str2=c.getText();
String str3=d.getText();
String str4=e.getText();
JOptionPane.showMessageDialog(this, “ “+ str+” “+str1+”class”+”
“+str2+”Roll no.”+str3+”%” “+str4+”CGPA”);
}
}

public static void main(String [] rag)
{
    new school(“hi”);
}
}

```

220. Write an applet using thread to give marquee effect in text.

```

import java.applet.*;
import java.awt.*;
import java.applet.*;
import java.awt.*;
public class Marquee extends Applet implements Runnable
{
String txt;
Image img;

```

```
Font f;
Color color;
int fontsize,fontstyle,pause;
Image offscreenimage;
Graphics ofg;
Thread mythread=null;
int offset=0;
String fontname;
public void init()
{
    offscreenimage=createImage(1000,1000);
    img=getImage(getDocumentBase(),"zoomimgp.jpg");
    ofg=offscreenimage.getGraphics();
    color=matchcolor(getParameter("color"));
    if(getParameter("fontsize")==null)
        fontsize=12;
    else
    {
        Integer i1=new Integer(getParameter("fontsize"));
        fontsize=i1.intValue();
    }
    fontname=getParameter("fontname");
    if(fontname==null)
        fontname="TimesRoman";
    fontstyle=get(getParameter("fontstyle"));
    if(getParameter("pause")!=null)
    {
        Integer i2=new Integer(getParameter("pause"));
        pause=i2.intValue();
    }
    else
        pause=100;
    f=new Font(fontname,fontstyle,fontsize);
    mythread=new Thread(this);
    txt=getParameter("disptxt");
```



```
if(txt==null)
{
txt="swati computers";
}
txt=txt+" ";
}
public void start()
{
if(mythread!=null)
{
mythread.start();
}
}
public void stop()
{
if(mythread!=null)
{
mythread.stop();
}
}
public int get(String str)
{
if(str==null)
return Font.PLAIN;
if(str.equalsIgnoreCase("plain"))
return Font.PLAIN;
if(str.equalsIgnoreCase("bold"))
return Font.BOLD;
if(str.equalsIgnoreCase("italic"))
return Font.ITALIC;
return Font.PLAIN;
}
public Color matchcolor(String str)
{
if(str==null)
```

```
return Color.black;
if(str.equalsIgnoreCase("red"))
return Color.red;
if(str.equalsIgnoreCase("orange"))
return Color.orange;
if(str.equalsIgnoreCase("cyan"))
return Color.cyan;
return Color.black;
}
public void paint(Graphics g)
{
String str;
if(offset==(int)txt.length())
{
offset=0;
}
else
{
offset++;
}
if(offset==0)
{
str=txt;
}
else
{
str=txt.substring(offset)+txt.substring(0,offset-1);
}
ofg.setFont(f);
ofg.setColor(color);
ofg.drawImage(img,0,0,this);
ofg.drawString(str,50,60);
g.drawImage(offscreenimage,0,0,this);
ofg.clearRect(0,0,this.size().width,this.size().height);
}
```

```
public void update(Graphics g)
{
    paint(g);
}
public void run()
{
    while(true)
    {
        repaint();
        try
        {
            mythread.sleep(pause);
        }
        catch(InterruptedException e)
        {
            System.out.println(e);
        }
    }
}
```

221. Write a program to input age and print message using assert.

```
import java.util.Scanner;

class Asssertion{
    public static void main( String args[] ){

        Scanner scanner = new Scanner( System.in );
        System.out.print("Enter ur age ");

        int value = scanner.nextInt();
        assert value>=18:"error";
        System.out.println("ok");

        System.out.println("value is "+value);
    }
}
```

```
}  
}
```

Write a program to select more than one item from list and generate bill.

```
import java.applet.Applet;  
import java.awt.*;  
import java.awt.event.*;  
/* <applet code="selecteditem.class" height="500" width="500"> </  
applet> */
```

```
public class selecteditem extends Applet implements  
ItemListener, ActionListener  
{
```

```
TextField t,t1;  
List l1;  
Button b;  
String p,q,r;  
int c;  
int d;  
Label l,l2;  
public void init()  
{
```

```
l=new Label("Guest");  
l2=new Label("Bill number");  
t=new TextField(10);  
t1=new TextField(10);  
add(l);  
add(t);  
add(l2);  
add(t1);  
b=new Button("order");  
b.addActionListener(this);
```

```
add(b);
l1=new List(2,true);
l1.add("pizza");
l1.add("burger");
l1.add("hotdog");
l1.add("footlong");
l1.add("donuts");
add(l1);
l1.addItemListener(this);
}
public void itemStateChanged(ItemEvent e)
{
repaint();
}
public void actionPerformed(ActionEvent ae)
{
System.out.println("\007");
r=t1.getText();
q=t.getText();
p=ae.getActionCommand();
if(p.equals("order"))
{
c=1;
repaint();
}
}
public void paint(Graphics g)
{
if(c==1)
{
g.drawString("Guest is"+q,150,430);
g.drawString("Bill no. is"+r,150,440);
g.drawString("your final bill is Rupees"+d,150,450);
g.drawString("Have a nice day!",150,460);
}
```

```
int i=0;

String val[];
String data="You have selected items";
val=l1.getSelectedItems();
int len=val.length;
while(len!=i)
{
    data=data+" "+val[i]+" ";
    String s="your expected bill is Rupees";
    int j=0;
    if(i==0)
    {j=j+10;
    g.drawString(s+j,150,170);
    d=j;
    }

    if(i==1)
    {j=j+20;
    g.drawString(s+j,150,180);
    d=j;
    }
    if(i==2)
    {j=j+30;
    g.drawString(s+j,150,190);
    d=j;
    }

    if(i==3)
    {j=j+40;
    g.drawString(s+j,150,200);
    d=j;
    }
    if(i==4)
    {j=j+50;
```

```

g.drawString(s+j,150,210);
d=j;
}
i++;
g.drawString(data,150,160);
}

}
}

```

222. Write a program to create digital clock without util.Calendar.

```

import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;

/*<applet code="tim" Height=700 Width=500></applet>*/
public class tim extends Applet implements Runnable
{
    TextField t0,t1,t2;
    Thread t;
    int hrs,min,sec;
    Label l1,l2,l3;
    String s;
    public void init()
    {
        t=new Thread(this);
        hrs=0;
        min=0;
        sec=0;

        t0=new TextField(2);
        t1=new TextField(2);
        t2=new TextField(2);
    }
}

```

```
t0.setEditable(false);
t1.setEditable(false);
t2.setEditable(false);
l1=new Label("hrs");
l2=new Label("min");
l3=new Label("sec");
add(l1);
add(t0);
add(l2);
add(t1);
add(l3);
add(t2);
t0.setText(" "+hrs);
t1.setText(" "+min);
t.start();
```

```
}
```

```
public void run()
{
while(true)
{
try
{

if(sec>=60)
{
min++;
t1.setText(" "+min);
sec=0;
if(min>=60)
{hrs++;
t0.setText(" "+hrs);
min=0;
if(hrs>=24)
```



```

    {hrs=0;}
  }

  }
  else
  {
    sec++;
    Thread.sleep(1000);
    t2.setText(" "+sec);
  }
}
catch(Exception e)
{}
finally
{}
}
}
}

```

223. Write a servlet to count how many times a user clicks (hits) on a button.

Hithtml.html

```

<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
  </head>
  <body>
    <form action=".\\hitserv" method="get">
      <input type="submit" value="Hit Me">
    </form>

    </body>
</html>

```

hitserv.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class hitserv extends HttpServlet {
    int hits;
    public void init()
    {
        hits=0;
    }
    public void doGet(HttpServletRequest req,HttpServletResponse res)
    throws IOException,ServletException
    {
        res.setContentType("text/html");
        PrintWriter pw=res.getWriter();
        RequestDispatcher rd=req.getRequestDispatcher("hithtml.html");
        hits++;
        pw.println("You hit me "+ hits + " times");
        rd.include(req, res);
    }

}
```

224. Write a program to show difference between String and StringBuilder class?

```
public class strng {
    public static void main(String [] ar)
    {
        String n=new String("Swati");
```

```

    System.out.println(n.hashCode());
    n="Computers";
    System.out.println(n.hashCode());
    n=n+"cloud";
    System.out.println(n.hashCode());
    StringBuilder sb=new StringBuilder("Swati");
    System.out.println(sb.hashCode());
    sb.append("Computers");
    System.out.println(sb);
    System.out.println(sb.hashCode());
}
}

```

225. Write a Java program to read an existing file with a try with resource.

Import java.io.*;

```

public class try_resource{

    public static void main(String[] args) {

        try(BufferedReader br=new BufferedReader(new FileReader
        ("C:\\testing.txt")))
        {
            String line;

            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }

        } catch (IOException e) {
            e.printStackTrace();
        }

    }
}

```

226. Write a program to display difference between equals() and == .

```
public class strng {
    public static void main(String [] ar)
    {
        String str1= new String("ABCD");
        String str2= new String("ABCD");
        if(str1 == str2)
        {
            System.out.println("String 1 == String 2 is true");
        }
        else
        {
            System.out.println("String 1 == String 2 is false");
            String Str3 = str2;
            if( str2 == Str3)
            {
                System.out.println("String 2 == String 3 is true");
            }
            else
            {
                System.out.println("String 2 == String 3 is false");
            }
            if(str1.equals(str2))
            {
                System.out.println("String 1 equals string 2 is true");
            }
            else
            {
                System.out.println("String 1 equals string 2 is false");
            }
        }
    }
}
```

SOME INTERVIEW QUESTION BASED ON PRACTICAL AND THEORY:

1. Explain public static void main(String[] arg).

Answer:

public : Public is an access modifier, which is used to specify who can access this method. Public means that this Method will be accessible by any Class.

static : It is a keyword in Java which identifies that it is class based i.e it can be accessed without creating the instance of a Class.

void : It is the return type of the method. Void defines the method which will not return any value.

main: It is the name of the method which is searched by JVM as a starting point for an application with a particular signature only. It is the method where the main execution occurs.

String args[] : It is the parameter passed to the main method.

2. Can public static void main(String[] arg) overload?

Answer:

Yes. But JVM search for main(String[] arg)

3. Why Java is not 100% Object-oriented?

Answer:

Java is not 100% Object-oriented because it makes use of eight primitive datatypes such as boolean, byte, char, int, float, double, long, short which are not objects.

4. What is singleton class and how can we make a class singleton?

Answer:

Singleton class is a class whose only one instance can be created at any given time, in one JVM. A class can be made singleton by making its constructor private.

5. Why we cannot override static method?

Answer:

It is because the static method is the part of class and it is bound with class whereas instance method is bound with object and static gets memory in class area and instance gets memory in heap.

6. Can you declare the main method as final?

Answer:

Yes, such as, `public static final void main(String[] args){}`.

7. Which class is the superclass for every class ?

Answer:

Object class

8. What is “this” in Java?

Answer:

“This” is a reference variable that refers to the current object.

9. What is static variable?

Answer:

static variable is used to refer the common property of all the objects (that is not unique for each object) .
static variable gets memory only once in class area at the time of class loading.

10. Which class is the superclass for every class.

Answer:

Object class.

11. Can we override static method?

Answer:

No, you can't override the static method because they are the part of class not object.

12. Define Path and ClassPath.

Answer:

Path specifies the location of .exe files while classpath is used for specifying the location of class files.

13. What is the difference between abstract class and interface?

Answer:

- 1) An abstract class can have abstract and non-abstract methods. An interface can only have abstract methods.
- 2) An abstract class can have static methods but an interface cannot have static methods.
- 3) An abstract class can have constructors but an interface cannot have constructors.

14. Why Strings in Java are called as Immutable?

Answer:

In Java, string objects are called immutable as once the value has been assigned to a string, it can't be changed and if changed, a new object is created.

15. How garbage collection is done in Java?

Answer:

In Java, when an object is not referenced any more, garbage collection takes place and the object is destroyed automatically. For automatic garbage collection Java calls either `System.gc()` method or `Runtime.gc()` method.

16. What is a servlet?

Answer:

Java Servlet is a server side technology to extend the capability of web servers by providing the support for dynamic response and data persistence.

The `javax.servlet` and `javax.servlet.http` packages provide interfaces and classes for writing our own servlets.

All servlets must implement the `javax.servlet.Servlet` interface, which defines servlet lifecycle methods.

When implementing a generic service, we can extend the `GenericServlet` class provided with the Java Servlet API.

The `HttpServlet` class provides methods, such as `doGet()` and `doPost()`, for handling HTTP-specific services.

Most of the times, web applications are accessed using HTTP protocol and that's why we mostly extend `HttpServlet` class.

17. What is Request Dispatcher?

Answer:

`RequestDispatcher` interface is used to forward the request to another resource that can be HTML, JSP or another servlet in same application. We can also use this to include the content of another resource to the response.

There are two methods defined in this interface:

1. `void forward()`
2. `void include()`

18. What are the steps to connect to a database in Java?

Answer:

- Registering the driver class
- Creating connection
- Creating statement
- Executing queries
- Closing connection

19. Can we have a try without catch block?

Answer:

Yes, we can have try-finally statement and hence avoiding catch block.

20. Java Compiler is stored in JDK, JRE, or JVM?

Answer:

The task of Java compiler is to convert Java program into bytecode, we have javac executable for that. So, it must be stored in JDK.

21. What is the difference between throw and throws in Java?

Answer:

Throws clause is used to declare an exception, which means it works similar to the try-catch block. On the other hand throw keyword is used to throw an exception explicitly.

22. Which types of exceptions are caught at compile time?

Answer:

Checked exceptions can be caught at the time of program compilation.
Checked exceptions must be handled by using try catch block or throws clause in the code in order to successfully compile the code.

23. In thread programming, which method must be implement by all threads?

Answer:

Run() is a method of Runnable interface or Thread class that must be implemented by all the threads.

24. What's the base class of all exception classes?

Answer:

In Java, `java.lang.Throwable` is the super class of all exception classes.

25. What is the difference between `GenericServlet` and `HttpServlet`?

Answer:

`GenericServlet` is protocol independent implementation of `Servlet` interface whereas `HttpServlet` is HTTP protocol specific implementation.

`HttpServlet` class extends `GenericServlet` and also provide some other methods specific to HTTP protocol.

26. What is a deployment descriptor?

Answer:

Deployment descriptor is a configuration file for the web application and it's name is `web.xml` and it resides in `WEB-INF` directory.

Servlet container use this file to configure web application servlets, servlet config params, context init params, filters, listeners, welcome pages, and error handlers.

27. What are the different methods of session management in servlets?

Answer:

Some of the common ways of session management in servlets are:
User Authentication
HTML Hidden Field

Cookies
URL Rewriting
Session Management API

28. What is Life cycle of a servlet?

Answer:

Following are the stages of servlet life cycle:

- 1) Loading of Servlet class: The servlet container finds the servlet class mentioned in web.xml file and loads it.
- 2) Servlet instantiation: The object of servlet class gets created in this phase.
- 3) Initialization: Servlet initialization by calling `init()` method.
- 4) Servicing the request: In this phase of the servlet service the client request by calling the `service()` method.
- 5) Destroy: Last phase of servlet life cycle. The `destroy()` method free up the servlet instance so that it can be garbage collected.

29. Why Servlet is better than CGI?

Answer:

Servlet responses faster than CGI because it uses the multithreading concept to service each request.

CGI performance is not that good as it creates a new object for each request while servlet allots a new thread for each request.

Memory consumption is low in servlet as compared to CGI.

30. How can we create deadlock condition on our servlet?

Answer:

one simple way to call `doPost()` method inside `doGet()` and `doGet()` method inside

`doPost()` it will create deadlock situation for a servlet.

31. What is the difference between the RMI and CORBA?

Answer:

RMI is a distributed object system that provides and enable to easily develop the distributed Java applications.

It is easier to develop applications using the RMI method then using the sockets. It doesn't require any protocol for design that makes it less prone to error for the tasks performed by it.

RMI allows the local method to be called from a local class file and the remote methods are interpreted and are sent back to the callers.

The CORBA also have the same features that are seen in RMI. It is a platform and language independent architecture that can be run on any platform.

It can be located from anywhere on the network and can be used in any language that has a mapping with **Interface definition language (IDL)**.

The objects of this are specified with interfaces that are specified in the interface language.

32. What are the advantages of using PreparedStatement in Java?

Answer:

Prepared Statement is used to execute same SQL statements repeatedly. The prepared statement is compiled only once even though it used "n" number of times.

33. What is the return type of execute, executeQuery, and executeUpdate?

Return type of execute is Boolean

Return type of executeQuery is ResultSet object

Return type of executeUpdate is int

34. What is JSP?

Answer:

JSPs are normal HTML pages with embedded Java code. To process a JSP file, developers need a JSP engine, which is connected

to a Web server.

The JSP page is then compiled into a servlet, which is handled by the servlet engine. This phase is known as translation.

The servlet engine then loads the servlet class and executes it to create dynamic HTML, which is then sent to the browser.

JSP was developed by Sun Microsystems and is an improved version of Java servlets.

As with most server-based technologies, JSP separates business logic from the presentation layer.

35. Implicit objects in JSP?

Answer:

1. response
2. exception
3. application
4. request
5. session
6. page
7. out
8. config
9. pageContext

36. JSP life cycle phase.

Answer:

- Translation: JSP container verifies the JSP page code and parses it to generate the servlet source code. For your note, if the JSP page name is <Login.jsp>, the generated servlet class name would be Login_jsp, and the Java file name would be Login_jsp.java.
- Compilation: JSP container compiles the JSP page and creates a class file in this phase.
- Class Loading: At this stage, the container reads the JSP class into application memory.
- Instantiation: This phase guides the container to execute the no-args constructor of the generated JSP class and loads it into memory to instantiate it.

- Initialization: Container calls the JSP class init method and initializes the servlet config with the parameters specified in the deployment descriptor. This phase makes sure that the JSP is fit to respond to the client requests.
- Request Processing: This is the phase where the JSP page create threads to process a request. Every new thread leads to the creation of ServletRequest and ServletResponse object followed by a call to the JSP service method.
- Destroy: This is the terminal phase of JSP life cycle where the JSP class gets unloaded from the memory. This stage comes in the picture when the application is undeployed, or the server is shut down.

37. JSP lifecycle methods?

Answer:

The JSP container executes the `jspInit()` method the first time it initializes the JSP.

Whenever the container receives a request, it invokes the `_jspService()` method, processes the request and generates a response.

The container calls the `jspDestroy()` method for cleaning up the memory allocated for the JSP.

38. What are the different JSP scripting elements?

Answer:

There are three types of scripting language elements:

Declarations

Scriptlets

Expressions

39. Is JSP technology extensible?

Answer:

YES. JSP technology is indeed extensible as it allows to create custom actions or tags for a large no. of use cases. JSTL libraries enable this functionality in a JSP page.

40. What is the difference between `<jsp:include page = ... >` and `<%@ include file = ... >?`

Answer:

Both the tag helps to include the information from one page to another:

`<jsp:include page = ... >`

This is a kind of function call from one jsp to another jsp.

It is executed whenever the client accesses the client page.

This approach is useful for modularizing the web applications.

`<%@ include file = ... >:`

The content of the included file is textually embedded in the page that has `<%@ include file="..">` directive.

By this the included file will be changed and the changed content can not be included in the output.

41. What are the difference between GET and POST method of HTML form?

Answer:

1. Visibility: GET request is sent via the URL string which is visible whereas POST request can't be seen as it is encapsulated in the body of the HTTP request.
2. Length : There is a limitation of length for GET request as it goes through URL and its character length cannot be more than 255 but in POST request, no such limitation.
3. Performance: As no time is spent on encapsulation in GET request so it is comparatively faster and relatively simpler. In addition, the maximum length restriction facilitates better optimization of GET implementation.
4. Type of Data: GET request can carry only text data as we know URL only contain text data on the other hand POST request can carry both text as well as binary.
5. Caching/Bookmarking: As we know, now that GET request is nothing but a URL hence it can be cached as well as

bookmarked. No such options are available with a POST request.

6. FORM Default: GET is the default method of the HTML FORM element. To submit a FORM using POST method, we need to specify the method attribute and give it the value 'POST'.
7. Data Set : GET requests are restricted to use ASCII characters only whereas no such restrictions are in POST requests.

42. Which is better from Extending Thread class and implementing Runnable Interface?

Answer:

Implementing runnable interface is more advantageous because when you are going for multiple inheritance, then only interface can help.

If you are already inheriting a different class, then you have to go for Runnable Interface. Otherwise you can extend Thread class.

If we implement runnable interface, it will provide better object oriented design.

Implementing also gives consistency.

43. Write the difference between Java array vs. ArrayList class?

Answer:

ArrayList is a dynamic array that can grow depending on demand whereas Java arrays are of fixed length.

44. What are the daemon threads?

Answer:

Daemon threads are designed to run in background. An example of such thread is a garbage collector thread.

45. What is a transient variable?

Answer:

Transient variable is a variable that may not be serialized.

46. What is Serialization?

Answer:

Serialization in Java is the process of converting an object into bytes stream to save it in file. Or we can say that, serialization is used to persist (save) the state of an object.

The reverse process of serialization is called deserialization.

47. What is the difference between TCP and UDP?

Answer:

These two protocols differ in the way they carry out the action of communicating.

A TCP protocol establishes a two way connection between a pair of computers, while the UDP protocol is a one way message sender. TCP is reliable protocol and UDP is unreliable.

48. What are the two important TCP Socket classes?

Answer:

Socket and ServerSocket.

ServerSocket is used for normal two way socket communication. Socket class allows us to read and write through the sockets. `getInputStream()` and `getOutputStream()` are the two methods available in Socket class.

49. What are the advantages of Java Sockets?

Answer:

Sockets are flexible and sufficient.

Efficient socket based programming can be easily implemented for general communications. It cause low network traffic.

50. What is `InetAddress` class?

Answer:

Java `InetAddress` class represents an IP address.

The `java.net.InetAddress` class provides methods to get the IP of any host name for example, `www.google.com`, `www.facebook.com`, and so on.

51. What is multiple inheritance and does Java support?

Answer:

If a child class inherits the property from multiple classes, it is known as multiple inheritance. Java does not allow to extend multiple classes.

The problem with the multiple inheritance is that if multiple parent classes have methods with same name, then at runtime it becomes difficult for the compiler to decide which method to execute from the child class.

To overcome this problem Java allows to implement multiple Interfaces. The problem is commonly referred as Diamond Problem.

52. What are class variable and instance variable?

Answer:

Class variables: belong to the class as a whole; there is only one copy of each one.

Instance variables or attributes - data that belongs to individual objects; every object has its own copy of each one.

53. What are the ways to overload method?

Answer:

A method can be overloaded by:

Varying the numbers of parameters

Using different data types for the parameters

Using different sequence of the parameters

54. Define a constructor?

Answer:

A constructor is a method used to initialize the state of an object, and it gets invoked at the time of object creation. Rules for constructor are:

Constructor Name should be same as class name.

x` A constructor must have no return type.

Constructor can be overload.

You need not to call it explicitly.

55. What is the use of finalize method?

Answer:

Finalize method helps to perform cleanup operations on the resources which are not currently used.

Finalize method is protected, and it is accessible only through this class or by a derived class.

56. What is the super keyword?

Answer:

Super keyword is used to invoke the overridden method which overrides one of its superclass methods.

Super keyword allows to access overridden methods and also to access hidden members of the superclass.

It also use to call super (base) class constructor.

57. What is a pointer and does Java support pointers?

Answer:

Pointer is a reference handle to a memory location. Improper handling of pointers leads to memory leaks and reliability issues hence Java doesn't support the usage of pointers.

58. Can a source file contain more than one class declaration?

Answer:

Yes, a single source file can contain any number of Class declarations but only one of the class can be declared as public.

59. If a class is declared without any access modifiers, where may the class be accessed?

Answer:

A class that is declared without any access modifiers is said to have package access. This means that the class can only be accessed by other classes and interfaces that are defined within the same package.

60. How to make a field not to serialize?

Answer:

There are three exceptions in which serialization does not necessarily read and write to the stream. These are as follows:

1. Serialization ignores static fields, because they are not a part of any particular state.
2. Base class fields are only handled if the base class itself is serializable.
3. Transient fields.

61. Why do we need wrapper classes?

Answer:

It is sometimes easier to deal with primitives as objects.

Moreover, most of the collection classes store objects and not primitive data types. And also the wrapper classes provide many utility methods also.

62. What is the difference between error and an exception?

Answer:

An error is an irrecoverable condition occurring at runtime. Such as OutOfMemory error.

These are JVM errors and you can not repair them at runtime.

While exceptions are conditions that occur because of bad input and exception can be handled by try-catch statements.

63. What is synchronization and why is it important?

Answer:

With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources.

Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often leads to significant errors.

64. What is the difference between sleep and wait in Java?

Answer:

Both are used to pause thread that is currently running, however, sleep() is meant for short pause because it does not release the lock, while wait() is meant for the conditional wait.

This is why it releases the lock, which can then be developed by a different thread to alter the condition of which it is waiting.

65. Explain the difference between JRE, JDK, JVM, and JIT.

Answer:

JRE is an abbreviation of Java Runtime Environment that consist of sets of files needed by JVM throughout the runtime.

JVM is an abbreviation of Java Virtual Machine which delivers the runtime environment for collected Java Bytecode. JVM is in

control of the conversion of the bytecode into machine-readable code.

JDK is an abbreviation for Java Development Kit which contains JRE including development tools for the purpose of development. JDK is required to write and execute a Java code.

JIT is an abbreviation of Just in Time compilation, and this helps to improve the performance of Java application by converting Java bytecode into native code when they cross certain threshold, i.e. the mostly hot code is transformed into native code.

66. What is the difference between final, finalize, and finally?

Answer:

Final is a modifier which you can apply to variable, methods, and classes. If you create a variable final, this means its value cannot be changed once initialised.

Finalise is a method, which is called just before an object is a garbage collected, allowing it a final chance to save itself, but the call to finalise is not definite.

Finally is a keyword which is used in exception handling, along with try and catch. The finally block is always implemented regardless of whether an exception is thrown from try block or not.

67. What is the immediate superclass of the Applet class?

Answer: Panel.

68. What is the wait(), notify(), and notifyAll() methods?

Answer:

The wait(), notify(), and notifyAll() methods are used to provide an efficient way for threads to wait for a shared resource.

When a thread executes an object's wait() method, it enters the waiting state. It only enters the ready state after another thread invokes the object's notify() or notifyAll() methods.

69. What is the difference between the `paint()` and `repaint()` methods?

Answer:

The `paint()` method supports the painting via a `Graphics` object. The `repaint()` method is used to cause `paint()` to be invoked by the AWT painting thread.

70. What are three ways in which a thread can enter the waiting state?

Answer:

A thread can enter the waiting state by invoking its `sleep()` method, by blocking on I/O, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's `wait()` method. It can also enter the waiting state by invoking its (deprecated) `suspend()` method.

71. What are the differences between Swing and AWT?

Answer:

The AWT component is considered to be heavyweight while Swing component are considered lightweights. The Swing GUI has pluggable look and feel. The AWT is platform dependent i.e. the same GUI will look different on different platforms because they use native components e.g. a `Frame` will look different on Windows and Linux if you pick native components. On the other hand, Swing components are developed in Java and are platform dependent. It also provide consistent GUIs across platform e.g. if you run NetBeans which is created in Java on Linux or Windows you will see consistency in GUI.

72. What is the difference between a `Scrollbar` and a `ScrollPane`?

Answer:

A `Scrollbar` is just a `Component`, but not a `Container`. A `ScrollPane` is a `Container`. A `ScrollPane` handles its own events and performs its own scrolling.

73. Disadvantages of Java Sockets :

Answer:

- 1.Socket based communications allows only to send packets of raw data between applications.
- 2.Both the client-side and server-side have to provide mechanisms to make the data useful in any way.

74. Why Java is not pure object oriented language?

Answer:

Pure Object Oriented Language are fully Object Oriented Language which supports or have features which treats everything inside program as objects.

It doesn't support primitive datatype (like int, char, float, bool, and so on).

Java supports primitive datatype.

75. Why wrapper class is used?

Answer:

Wrapper class provides the mechanism to convert primitive into object and vice versa.

76. Brief history of Java?

Answer:

Java is a general purpose, high-level programming language developed by Sun Microsystems. The Java programming language was developed by a small team of engineers, known as the Green Team, who initiated the language in 1991.

The Java language was originally called OAK, and at the time it was designed for handheld devices and set-top boxes. Oak was unsuccessful and in 1995 Sun changed the name to Java and modified the language to take advantage of the World Wide Web.

Later in 2009, Oracle Corporation acquired Sun Microsystems and took ownership of two key Sun software assets: Java and Solaris.

77. What is JVM?

Answer:

JVM is a engine that provides runtime environment to drive the Java Code or applications.

It converts Java bytecode into machines language.

JVM is a part of **Java Run Environment (JRE)**. It stands for Java Virtual Machine.

78. What is classloader?

Answer:

The class loader is a subsystem used for loading class files. It performs three major functions viz. Loading, Linking, and Initialization.

79. What is Garbage collector?

Answer:

Garbage collection (GC) is the process that aims to free up occupied memory that is no longer referenced by any reachable Java object,

and is an essential part of the Java virtual machine's (JVM's) dynamic memory management system.

All Java objects automatically grab the memory that they need when they are created, and when the object is no longer need, the Java Garbage Collection process reclaim the memory.

That means, the Garbage Collector tracked live objects and everything else designated garbage.

80. What is transient variable in Java?

Answer:

Transient is a variables modifier used in serialization.

At the time of serialization, if we don't want to save value of a particular variable in a file, then we use transient keyword.

When JVM comes across transient keyword, it ignores original value of the variable.

81. What is synchronization?

Answer:

At times when more than one thread try to access a shared resource, we need to ensure that resource will be used by only one thread at a time.

The process by which this is achieved is called synchronization.

82. What is autoboxing and unboxing?

Answer:

Autoboxing and Unboxing features were added in Java5.

The automatic conversion of primitive data types into its equivalent Wrapper type is known as boxing and opposite operation is known as unboxing.

Converting a primitive value (an int, for example) into an object of the corresponding wrapper class (Integer) is called autoboxing.

The Java compiler applies autoboxing when a primitive value is:

1. Passed as a parameter to a method that expects an object of the corresponding wrapper class.
2. Assigned to a variable of the corresponding wrapper class.
 1. Autoboxing is a process by which primitive type is automatically encapsulated (boxed) into its equivalent type wrapper.
 2. Auto-Unboxing is a process by which the value of an object is automatically extracted from a type Wrapper class.

83. What are JDBC Driver?

Answer:

JDBC Driver is a software component that enables Java application to interact with the database. There are four types of

JDBC drivers:

1. JDBC-ODBC bridge driver
2. Native-API driver (partially Java driver)
3. Network Protocol driver (fully Java driver)
4. Thin driver (fully Java driver)

Type 1: JDBC-ODBC Bridge Driver

In a Type 1 driver, a JDBC bridge is used to access ODBC drivers installed on each client machine. Using ODBC, requires configuring on your system a **Data Source Name (DSN)** that represents the target database.

When Java first came out, this was a useful driver because most databases only supported ODBC access but now this type of driver is recommended only for experimental use or when no other alternative is available.

ODBC is an open standard **application programming interface (API)** for accessing a database.

By using ODBC statements in a program, you can access files in a number of different databases, including Access, dBase, DB2, Excel, and Text. In addition to the ODBC software, a separate module or driver is needed for each database to be accessed.

The main of ODBC programming support is Microsoft.

Type 2: JDBC-Native API

In a Type 2 driver, JDBC API calls are converted into native C/C++ API calls, which are unique to the database.

These drivers are typically provided by the database vendors and used in the same manner as the JDBC-ODBC Bridge. The vendor-specific driver must be installed on each client machine.

The Native API driver uses the client-side libraries of the database. The driver converts JDBC method calls into native calls of the database API. It is not written entirely in Java.

Type 3: Network Protocol driver / JDBC-Net pure Java

In a Type 3 driver, a three-tier approach is used to access databases. The JDBC clients use standard network sockets to communicate with a middleware application server.

The socket information is then translated by the middleware application server into the call format required by the DBMS, and forwarded to the database server.

Type 4: Fully Java Driver

This is the highest performance driver available for the database and is usually provided by the vendor itself.

This kind of driver is extremely flexible, you don't need to install special software on the client or server. Further, these drivers can be downloaded dynamically.

84. What is an API?

Answer:

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other.

Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an API.

For Example:

When you use an application on your mobile phone, the application connects to the Internet and sends data to a server.

The server then retrieves that data, interprets it, performs the necessary actions and sends it back to your phone.

The application then interprets that data and presents you with the information you wanted in a readable way. This is what an API is - all of this happens via API.

85. Why would it be more secure to store sensitive data (such as a password, social security number, and so on) in a character array rather than in a String?

Answer:

In Java, Strings are immutable and are stored in the String pool. What this means is that, once a String is created, it stays in the pool in memory until being garbage collected.

Therefore, even after you're done processing the string value (e.g., the password), it remains available in memory for an indeterminate period of time thereafter (again, until being garbage collected) which you have no real control over.

Therefore, anyone having access to a memory dump can potentially extract the sensitive data and exploit it.

In contrast, if you use a mutable object like a character array, for example, to store the value, you can set it to blank, once you are done with it with confidence that it will no longer be retained in memory.

86. SQL Query to find second highest salary of an Employee?

```
select MAX (Salary) from Employee WHERE Salary NOT IN
(select MAX(Salary) from Employee
```

87. Write an SQL Query to find number of employees according to gender whose DOB is between 01/01/1990 to 31/12/2010.

Answer:

```
SELECT COUNT(*), gender from Employees WHERE DOB
BETWEEN '01/01/1990' AND '31/12/2010' GROUP BY
gender;
```

88. Write an SQL Query to find duplicate rows in a database?

Answer:

```
SELECT * FROM emp a WHERE rowid = (SELECT
MAX(rowid) FROM EMP b WHERE a.empno=b.empno)
```

89. Explain some of the methods of Calendar class?

Answer:

```
Calendar calendar = Calendar.getInstance();
System.out.println(calendar.get(Calendar.YEAR));
System.out.println(calendar.get(Calendar.MONTH));
System.out.println(calendar.get(Calendar.DATE));
System.out.println(calendar.get(Calendar.WEEK_OF_
MONTH));
System.out.println(calendar.get(Calendar.WEEK_OF_YEAR));
System.out.println(calendar.get(Calendar.DAY_OF_YEAR));
System.out.println(calendar.getFirstDayOfWeek());//1    ->
Calendar.SUNDAY
```

90. What modifiers may be used with an inner class that is a member of an outer class?

Answer:

A inner class may be declared as public, protected, private, static, final, or abstract.

91. How are the JSP requests handled ?

Answer:

On the arrival of a JSP request, the browser first requests a page with a .jsp extension. Then, the Web server reads the request and using the JSP compiler,

the Web server converts the JSP page into a servlet class.

Notice that the JSP file is compiled only on the first request of the page, or if the JSP file has changed. The generated servlet class is invoked, in order to handle the browser's request.

Once the execution of the request is over, the servlet sends a response back to the client.

92. What are JSP actions ?

Answer:

JSP actions use constructs in XML syntax to control the behavior of the servlet engine. JSP actions are executed when a JSP page is requested.

They can be dynamically inserted into a file, re-use JavaBeans components, forward the user to another page, or generate HTML for the Java plugin. Some of the available actions are listed below:

- jsp:include – includes a file, when the JSP page is requested.
- jsp:useBean – finds or instantiates a JavaBean.
- jsp:setProperty – sets the property of a JavaBean.
- jsp:getProperty – gets the property of a JavaBean.
- jsp:forward – forwards the requester to a new page.
- jsp:plugin – generates browser-specific code.

93. Explain different ways of creating a thread. Which one would you prefer and why ?

Answer:

There are three ways that can be used in order for a Thread to be created:

A class may extend the Thread class.

A class may implement the Runnable interface.

An application can use the Executor framework, in order to create a thread pool.

The Runnable interface is preferred, as it does not require an object to inherit the Thread class.

In case your application design requires multiple inheritance, only interfaces can help you. Also, the thread pool is very efficient and can be implemented and used very easily.

94. What is the applet security manager, and what does it provide?

Answer:

The applet security manager is a mechanism to impose restrictions on Java applets. A browser may only have one security manager.

The security manager is established at startup, and it cannot thereafter be replaced, overloaded, overridden, or extended.

95. What is the difference between Exception and Error in Java ?

Answer:

Exception and Error classes are both subclasses of the Throwable class.

The Exception class is used for exceptional conditions that a user's program should catch.

The Error class defines exceptions that are not expected to be caught by the user program.

96. Explain the available thread states?

Answer:

During its execution, a thread can reside in one of the following states:

Runnable: A thread becomes ready to run, but does not necessarily start running immediately.

Running: The processor is actively executing the thread code.

Waiting: A thread is in a blocked state waiting for some external processing to finish.

Sleeping: The thread is forced to sleep.

Blocked on I/O: Waiting for an I/O operation to complete.

Blocked on Synchronization: Waiting to acquire a lock.

Dead: The thread has finished its execution.

97. How do you ensure that N threads can access N resources without deadlock ?

Answer:

A very simple way to avoid deadlock while using N threads is to impose an ordering on the locks and force each thread to follow that ordering.

Thus, if all threads lock and unlock the mutexes in the same order, no deadlocks can arise.

98. What is the output of following program?

```
public class Maths {
    public static void main(String [] ar)
    {
        int x = 10*10-10;

        System.out.println(x);
    }
}
```

Answer: 90

99. Write a program to explain array having different columns in rows?

Answer:

```
public class Arrays {
    public static void main (String [] ar)
    {
        int [][]a=new int[][]{{1,2,3},{1,2},{1,2,3,4}};

        for(int i=0;i<a.length;i++)
        {
            System.out.print("Row" +(i+1) + "\t=\t");

            for(int j=0;j<a[i].length;j++)
            {
                System.out.print(a[i][j]);
            }

            System.out.println("");
        }
    }
}
```

100. Write a program to find duplicate (only first duplicate) value in a string?

Answer:

```
import java.util.*;
public class Duplicate {
```

```
public static void main(String [] ar)
{

    String [] sValue = new String[]{“abc”,”bcd”,”asd”,”dca”,”bcd”};

    System.out.println(“NormalMethod\n”+FindDupValue(sValue));

    System.out.println(“Using Builtin Class Set \n”+FindDupValue
    UsingSet(sValue));

}

private static String FindDupValue(String[] sValueTemp)
{
    for (int i = 0; i < sValueTemp.length; i++) {
        String sValueToCheck = sValueTemp[i];
        if(sValueToCheck==null || sValueToCheck.equals(“”))continue;
        for (int j = 0; j < sValueTemp.length; j++) {
            if(i==j)continue;
            String sValueToCompare = sValueTemp[j];
            if (sValueToCheck.equals(sValueToCompare)){
                return sValueToCompare;
            }
        }
    }

    return “”;

}

private static String FindDupValueUsingSet(String[]
sValueTemp)
{
    Set sValueSet = new HashSet();
    for(String tempValueSet : sValueTemp)
    {
        if (sValueSet.contains(tempValueSet))
```

```

return tempValueSet;
else
if(!tempValueSet.equals(""))
sValueSet.add(tempValueSet);
}
return "";
}
}

```

101. Where to write return statement?

Answer:

i) Return statement in try block only

If we return a value in try block and if we do not return a value at the end of the method it leads to compile time exception.

Error: This method must return a result of type int.

```
public class TryCatchReturn{
```

```
int fun(){ // Error:This method must return a result of type int
```

```
try {
```

```
return 1;
```

```
} catch (Exception e) {
```

```
}
```

```
System.out.println("End of the method");
```

```
}
```

```
public static void main(String[] args) {
```

```
TryCatchReturn obj = new TryCatchReturn();
```

```
}
```

```
}
```

ii) If a method returns any value you can write “return” statement in try block but after return statement if you write any code, it will be unreachable code and the compiler will raise an error.

```
public class TryCatchReturn{
```

```
int fun(){
```

```
try {
```

```
return 1;
```

```
} catch (Exception e) {
```

```
}
```

```
return 10;
```

```
System.out.println(“End of the method”); // Error : Unreachable  
code  
}
```

```
public static void main(String[] args) {
```

```
TryCatchReturn obj = new TryCatchReturn();
```

```
}
```

```
}
```

correct:

iii) return statement in try block and end of the method

102. Write a program to print Pascals Triangle using Java code?

Answer:

```
public class Pascals {
    public static void main(String [] ar)
    {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter number of rows ");

        int rows= in.nextInt();

        for(int i =0;i<rows;i++) {

            int number = 1;

            System.out.format("%"+(rows-i)*2+"s", "");

            for(int j=0;j<=i;j++) {
                System.out.format("%4d",number);

                number = number * (i - j) / (j + 1);

            }

            System.out.println();

        }
    }
}
```

103. Write a program to print semicolon without using semicolon?

Answer:

```
public class SemiColon{
    public static void main(String [] ar)
    {

        if(System.out.printf("%C",59) != null){}

    }
}
```

104. What are the five different ways to print array?

Answer:

```
public class Arys {  
    public static void main(String [] ar)  
    {  
        int[] narray = { 12,13,8,34,2,7,9,43,54,21};
```

```
//1.
```

```
for (int i = 0; i < narray.length; i++) {  
    System.out.println(narray[i]);  
}
```

```
//2.
```

```
` String[] array1 = { "Java", "At", "SwatiComputers"};
```

```
for(String st:array1)  
    System.out.println(st);
```

```
//3.
```

```
System.out.println(Arrays.toString(array1));
```

```
int[][] array = new int[][]{  
    {1,2,3},  
    {11,12,13},  
    {4 ,5,6},  
};
```

```
//4.
```

```
System.out.println(Arrays.deepToString(array));  
String[] strarray={"hi","array","print"};
```

```
//5.
```

```
System.out.println(Arrays.asList(strarray));  
}
```

105. Write a program to open Notepad using Java code?

Answer:

```
public class Notep {  
    public static void main(String [] ar)  
    {  
        Runtime rt = Runtime.getRuntime();  
  
        try {  
            rt.exec("notepad");  
        }  
        catch (Exception ex) {  
  
            System.out.println(ex);  
  
        }  
    }  
}
```

Advance Java New Updates

1. Explain stream API in java 8 ?

A stream is a sequence of elements from collection or array, that support various methods which can be pipelined to produce desired result.

For example you watched online videos on Youtube. When we start watching a video, a small portion of the file is first loaded into the computer and starts playing. we don't need to download the complete video before we start playing it. This is called streaming.

We can think of that small portions of the video file as a stream, and the whole video as a Collection.

In Java 8 stream API is used to process collection of objects.

Example to create stream –

```
import java.util.*;
import java.util.stream.Collectors;
import java.util.stream.Stream;

class strm
{
    public static void main(String [] er)
    {
        Stream<Integer> stream = Stream.of(1,2,3,4,5,6,7,8,9);

        stream.forEach(p -> System.out.println(p));

        // create stream from array

        Stream<Integer> stream1 = Stream.of( new Integer[]{1, 2, 3, 4, 5,
        6, 7, 8, 9} );

        stream1.forEach(p -> System.out.println(p));

        // create stream from list
```



```
List<Integer> list = new ArrayList<Integer>();

    for(int i = 1; i< 10; i++){
        list.add(i);
    }

    Stream<Integer> stream2 = list.stream();
    stream2.forEach(p -> System.out.println(p));
}
}
```

2. What is the difference between Java 8 and java 9 ?
 - A. Java 8 and earlier versions use “package” to store related class , interface etc while java 9 introduce a new component called “module” which can be used to place a set of related packages into a group, and also another new component: the **module descriptor, module-info.java** file.
 - B. Java 9 includes new features:

JShell, Http2Client, Java Platform Module System (JPMS), Multi-release jar files, Stack Walking API, Private methods in an interface, Process API updates, Collection API updates, Stream API improvements, and etc.

3. Explain takewhile () in java 9 with example?

This method returns a stream of the remaining elements of the given stream after taken the longest prefix of elements that match the given predicate if the stream is ordered else a stream of a subset of elements taken from this stream that match the given predicate.

```
import java.util.*;

import java.util.stream.Collectors;

import java.util.stream.Stream;
```

```
class switch
{
    public static void main(String [] er)
    {
        // create a stream of names
        Stream<String> stream
            = Stream.of("john", "joseph", "johhny", "scott",
                "joseph");

        // apply takeWhile to take all the names
        // matches passed predicate
        List<String> list
            = stream.takeWhile(name -> (name.charAt(0) == 'j'))
                .collect(Collectors.toList());

        // print list
        System.out.println(list);
    }
}

// create a stream of names
stream
    = Stream.of("john", "joseph", "adam", "scott", "joseph");
list
    = stream.takeWhile(name -> (name.charAt(0) == 'j'))
        .collect(Collectors.toList());

// print list
System.out.println(list);
```

```
G:\java Update>java swtch.java
[john, joseph, johhny]
[john, joseph]
```

4. What are the new features of Java 11?

Answer:

- Running Java File with single command
- New utility methods in String class
- Local-Variable Syntax for Lambda Parameters
- Nested Based Access Control
- JEP 321: HTTP Client
- Reading/Writing Strings to and from the Files
- JEP 328: Flight Recorder

5. How to compile and run java file with single command (java 12)?

Answer:

Suppose your java file name is "First.java"

C: \ \ > java First.java

```
G:\java Update>java First.java
```

6. Write a program to check whether user enter a blank string or his name?

Answer:

```
import java.util.*;

public class First
{
    public static void main(String [] as)
```

```
{  
String name;  
    Scanner sc=new Scanner(System.in);  
    System.out.println("Enter your name");  
    name=sc.nextLine();  
    if(name.isBlank())  
        System.out.println("Do not enter blank");  
    else  
        System.out.println("Welcome "+name);  
  
}  
}
```

7. Write a program to returns a stream of lines extracted from the string, separated by line terminators.

Answer:

```
import java.util.*;  
import java.util.stream.*;  
public class First  
{  
    public static void main(String [] as)  
    {  
        String str = "swati\ncomputers\nJaipur";  
        System.out.println(str.lines().collect(Collectors.toList()));  
        List<String> val=str.lines().collect(Collectors.toList());  
        System.out.println(val.get(0)+ " " +val.get(1)+" " +val.get(2));  
    }  
}
```

```
}
```

Output:

```
G:\java Update>java First.java
[swati, computers, Jaipur]
swati computers Jaipur
```

8. Write a program to enter a string and repeat it as many times user wants, without using any loop?

Answer:

```
import java.util.*;
import java.util.stream.*;
public class First
{
    public static void main(String [] as)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter String");
        String str=sc.nextLine();
        System.out.println("How many times you want to repeat
it?");
        int count=sc. nextInt ();
        str=str. repeat(count);
        System.out.println(str);
    }
}
```

```
G:\java Update>java First.java
Enter String
hello
How many times you want to repeat it?
10
hellohellohellohellohellohellohellohellohellohello
```

9. Write a program to remove whitespaces from start and end of string ?

```
import java.util.*;
import java.util.stream.*;
public class First
{
    public static void main(String [] as)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter String");
        String str=sc.nextLine();
        System.out.println("Before remove : "+str+"\n"+"After
        remove :"+str.strip());
    }
}
```

```
G:\java Update>java First.java
Enter String
    swati
Before remove :    swati
After remove :swati
```

10. Write a program to print following pattern using repeat() ?

```
*
*      *
*      *      *
*      *      *      *
*      *      *      *      *
```

```
import java.util.*;
import java.util.stream.*;
public class First
{
    public static void main(String [] as)
    {
        int i;
        String s="*";
        for(i=1;i<=5;i++)
        {
            System.out.println(s.repeat(i));
        }
    }
}
```

11. Write a program to multiply two given numbers using lambda function?

Answer:

```
import java.util.*;
import java.util.stream.*;
interface maths
```

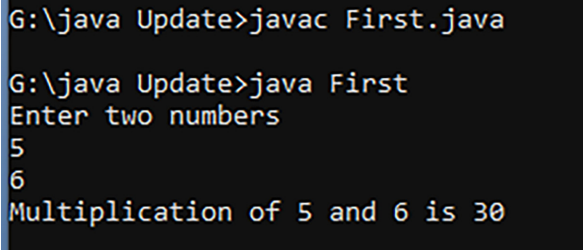
```
{
int multiply(int x, int y);
}

public class First
{
public static void main(String [] as)
{
Scanner sc=new Scanner(System.in);

    System.out.println("Enter two numbers");
    int a=sc. nextInt ();
    int b=sc. nextInt ();

    maths multi =(x,y)->(x*y);

    System.out.println("Multiplication of "+a+" and "+b + " is
    "+multi.multiply(a,b));}
}
```



```
G:\java Update>javac First.java
G:\java Update>java First
Enter two numbers
5
6
Multiplication of 5 and 6 is 30
```

12. Explain JEP 321: HTTP Client ?

The new API supports both HTTP / 1.1 and HTTP / 2. It is designed to improve the overall performance of sending requests by a client and receiving responses from the server.

13. Explain Nested Based access control ?

The term “nest “ defines a new access control context that allows classes that are logically part of the same code entity, but which are compiled with distinct class files, to access each other’s private members without the need for compilers to insert accessibility-broadening bridge methods .

However, Java 11 introduces the *nests* access control context, which provides support for `private` access within outer and nested classes

```
public class Second {

    public void myPublic() {

    }

    private void myPrivate() {

    }

    class Nested {

        public void nestedPublic() {
            myPrivate();
        }

    }

    public static void main(String [] ssd)
    {
        Second.Nested n=new Second().new Nested();
        n.nestedPublic();
    }
}
```

```
    }  
}
```

14. Explain java flight –recorder?

Answer:

JFR Provide a low-overhead data collection framework for troubleshooting Java applications and the HotSpot JVM.

JFR is a profiling tool used to gather diagnostics and profiling data from a running Java application.

Flight Recorder records events originating from applications, the JVM and the OS. Events are stored in a single file that can be attached to bug reports and examined by support engineers, allowing after-the-fact analysis of issues in the period leading up to a problem. Tools can use an API to extract information from recording files.

15. Write a program to check whether a given weekday is even or odd day using switch –case (switch expression).

Answer:

```
import java.util.*;  
class swtch  
{  
    public static void main(String [] er)  
    {  
        String day;  
        Scanner sc=new Scanner(System.in);  
        System.out.println("Enter an alphabet");  
        day=sc.next();  
        String res="";
```

```

switch (day) {
    case "M":
    case "W":
    case "F": {
        res = "odd";
        break;
    }
    case "T":
    case "TH":
    case "S": {
        res = "even";
        break;
    }
    default:
        res="sunday";
};System.out.println(res);
}
}

```

```

G:\java Update>java swtch.java
Enter an alphabet
Su
sunday

G:\java Update>java swtch.java
Enter an alphabet
W
odd

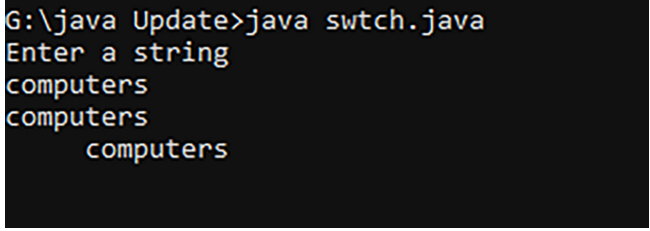
G:\java Update>java swtch.java
Enter an alphabet
S
even

```

16. Write a program to explain indent() method ?

```
import java.util.*;

class swtch
{
    public static void main(String [] er)
    {
        String day;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter any word");
        day=sc.next();
        System.out.println(day);
        System.out.println(day.indent(5));
    }
}
```



A screenshot of a terminal window with a black background and white text. The prompt is 'G:\java Update>'. The user enters 'java swtch.java'. The program outputs 'Enter a string', then 'computers' on the next line, and 'computers' on the third line. On the fourth line, the word 'computers' is indented by five spaces.

17. Write a program in java to swap two string without using any third variable ?

```
import java.util.*;

class swtch
{
    public static void main(String [] er)
    {
```

```

Scanner sc=new Scanner(System.in);

System.out.println("Enter two strings");

String s1 = sc.next();

String s2 = sc.next();


System.out.println(String.format("Before Swapping s1 = %s
and s2 = %s.\n", s1, s2));


s1 = s1 + s2; // s1 = "AppleBanana";

s2 = s1.substring(0, s1.length()-s2.length()); // s2 =
"AppleBanana".substring(0, 11-6) = "Apple"

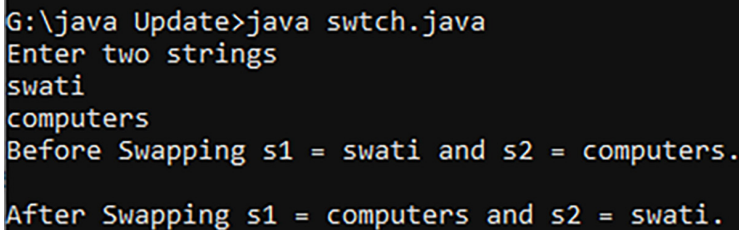
s1 = s1.substring(s2.length()); // s1 = "AppleBanana".
substring(5) = "Banana"


System.out.println(String.format("After Swapping s1 = %s
and s2 = %s.\n", s1, s2));

}

}

```



```

G:\java Update>java swtch.java
Enter two strings
swati
computers
Before Swapping s1 = swati and s2 = computers.
After Swapping s1 = computers and s2 = swati.

```

18. Explain switch expression in java14 ?

```

import java.util.*;

public class Main {

    public static void main(String[] args) throws Exception {

```

```
//Scanner sc=new Scanner(System.in);

//System.out.println("Enter an alphabet of weekday
name");

//String day=sc.next();

String day="S";

String result = switch (day) {
    case "M", "W", "F" -> "odd";
    case "T", "TH", "S" -> "even";
    default -> {
        if(day.isEmpty())
            yield "Please insert a valid day.";
        else
            yield "Looks like a Sunday.";
    }
};

System.out.println(result);

}
```

19. What is compact number formatting?

Java 12 introduces a convenient method called

NumberFormat.getCompactNumberInstance(Locale, NumberFormat.Style) for creating a compact number representation.

Suppose we need to parse a number 5,5,00,000.

Using compact number formatting we can read this easily as 5.5M .

Example:

```
import java.util.*;
import java.text.NumberFormat;
import java.util.Locale;

public class Main {

    public static void main(String[] args) throws Exception {

        NumberFormat formatter = NumberFormat.getCompactNumberInstance(Locale.US, NumberFormat.Style.SHORT);

        String formattedString = formatter.format(25000L);

        System.out.println(formattedString);

    }

}
```

Output : 25K

20. Give an example to set the minimum number of digits allowed in the fraction portion of a number using compact number formatting?

```
import java.util.*;
import java.text.NumberFormat;
import java.util.Locale;

public class Main {

    public static void main(String[] args) throws Exception {

        NumberFormat fmt = NumberFormat.getCompactNumberInstance(Locale.US, NumberFormat.Style.SHORT);

        fmt.setMinimumFractionDigits(3);

        System.out.println( fmt.format(10000) );

    }

}
```

```
        System.out.println( fmt.format(10012) );  
        System.out.println( fmt.format(100201) );  
        System.out.println( fmt.format(1111111) );  
  
    }  
}
```

Output: 10.000K

10.012K

100.201K

1.111M