Ex.No.	1	Implementation Of Electricity Bill Generation
Date		

AIM

To develop a java application for generating the Electricity Bill

ALGORITHM

Step 1: Start the program.

Step 2: Get the number units conceived from the user.

Step 3: Check units consumed is less than equal to the 100, If yes then the total electricity bill will be Rs.10/ Unit.

Step 4: Else if, check that unit consumed is less than equal to the 200, if yes then total electricity bill will be Rs. 15/ Unit.

Step 5: Else if, check that unit consumed is less than equal to the 300, if yes then total electricity bill will be Rs. 20/ unit.

Step 6: Else if, check that unit consumed greater than 300, if yes then total electricity bill will be Rs. 25/ unit.

Step 7: Execute the program.

Step 8: Stop the program.

PROGRAM:

```
import java.util.*;
class ElectricityGenerator
{
        public static void main(String args[])
        {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number of unites conceived");
        int result;
        int U=sc.nextInt();
        if(U>=1)
        {
                if((U>=1)&(U<=100))
                {
                        result=U*10;
                        System.out.println("Electricity Bill= "+result);
                }
                else if((U>100)&(U<=200))
                {
                        result=U*15;
                        System.out.println("Electricity Bill= "+result);
                }
                else if((U>200)&(U<=300))
                {
                        result=U*20;
                        System.out.println("Electricity Bill= "+result);
                }
                else if(U>300)
                {
                        result=U*25;
                        System.out.println("Electricity Bill= "+result);
                }

        }
        else
                {
                        System.out.println("Electricity Bill is NILL");
                }
}
}
```

OUTPUT:

```
E:\PJ>javac ElectricityGenerator.java
E:\PJ>java ElectricityGenerator
Enter the number of unites conceived
102
Electricity Bill= 1530
```

INFERENCE:

VIVA QUESTIONS:

1. Define class.

2. What are the OOP Principles?

3. Explain about scanner.

4. Write about else if ladder.

5. Explain about logical & operator.

Ex.No.	2	IMPLEMENTATION OF PAYROLL SYSTEM
Date		

AIM:

To develop a java Program to calculate the net salary of an employee.

ALGORITHM:

Step 1: Start the program.

Step 2: Create the methods to get the information, to calculate the salary and also to display the information.

Step 3: Get the basic salary from the employee.

Step 4: Calculate the House Rent Allowance (HRA) as 10% of the basic salary.

Step 5: Calculate the Dearness Allowance (DA) as 73% of basic salary.

Step 6: Calculate the Gross salary (GS) as the added value of basic salary, DA and HRA.

Step 7: Calculate the income tax as 30% of gross salary.

Step 8: Calculate the net salary as the subtracted value of GS and income tax.

Step 9: Execute the program.

Step 10: Stop the program.

PROGRAM:

```
import java.util.*;
class empinfo
{
	String name,id;
	int bp;
	float HRA,DA,GS,income_tax,net_salary;
	Scanner sc=new Scanner(System.in);
	public void getinfo()
	{
		System.out.println("Enter the Employee ID: ");
		id=sc.next();
		System.out.println("Enter the Employee Name: ");
		name=sc.next();
		System.out.println("Enter the basic salary of an Employee: ");
		bp=sc.nextInt();
	}
	public void calculate()
	{
		System.out.println(bp);
		HRA=bp*0.1f;
		DA=0.73f*bp;
		GS=bp+DA+HRA;
		income_tax=0.3f*GS;
		net_salary=GS-income_tax;
	}
	public void display()
	{
		System.out.println("HRA= "+HRA);
		System.out.println("DA= "+DA);
		System.out.println("GS= "+GS);
		System.out.println("INCOME TAX= "+income_tax);
		System.out.println("NET SALARY= "+net_salary);
	}
}
public class emp
{
	public static void main(String args[])
	{
		empinfo e=new empinfo();
		e.getinfo();
		e.calculate();
		e.display();
	}
}
```

OUTPUT:

```
E:\PJ>javac emp.java
E:\PJ>java emp
Enter the Employee ID:
7893

Enter the Employee Name:
RAMU.K
Enter the basic salary of an Employee:
75000
75000
HRA= 7500.0
DA= 54750.0
GS= 137250.0
INCOME TAX= 41175.0
NET SALARY= 96075.0
```

INFERENCE:

VIVA QUESTIONS:

1. Define method in java

2. What is the usage of util package?

3. How to call methods in java

4. Explain main( ) method in java

5. What are all the primitive data types supported by Java

Ex.No.	3	IMPLEMENTATION OF CURRENCY CONVERSION AND CURRENCY EXCHANGE
Date		

AIM

To develop a java Program for Currency Conversion and Currency Exchange by generating a constructor.

ALGORITHM

Step 1: Start the program.

Step 2: Create the constructor for currency conversion.

Step 3: Get the input from the user for currency conversion.

Step 4: Convert US dollars to Euros using following formula Euros=US dollars*0.82

Step 5: Convert US dollars to British pounds using following formula pounds=US dollars*0.70

Step 6: Convert US dollars to Japanese Yen using following formula Yen=US dollars*105.61

Step 7: Convert Euros to US dollars using following formula Dollars = Euros*1.22

Step 8: Convert Euros to British Pounds using following formula pounds = Euros*0.86

Step 9: Convert Euros to Japanese Yen using following formula Yen = Euros*128.54

Step 11: Convert British Pounds to dollars using following formula dollars= pounds*1.42

Step 12: Convert British Pounds to Euros using following formula Euros= pounds*1.17

Step 13: Convert British Pounds to Japanese Yen using following formula Yen= pounds*149.75

Step 14: Convert Japanese Yen to US Dollars using following formula dollars= Yen*0.0095

Step 15: Convert Japanese Yen to Euros using following formula Euros= Yen*0.0078

Step 16: Convert Japanese Yen to British pounds using following formula pounds= Yen*0.00067

Step 17: Execute the program.

Step 18: Stop the program.

PROGRAM:

```
import java.util.Scanner;

public class currency
{
    public currency()
    {
        char us_dollar_sym = 36;
        char pound_sym = 163;
        char yen_sym = 165;
        char euro_sym = 8364;

        String us_dollar = "Dollars";
        String pound = "Pounds";
        String yen = "Yen";
        String euro = "Euros";
        double rate = 0;


        // Interface
        System.out.println("Welcome to the Currency Converter Program \n");
        System.out.println("Use the following codes to input your currency choices: \n 1 - US dollars \n 2 -
Euros \n 3 - British Pounds \n 4 - Japanese Yen \n");

        //
        System.out.println("Please choose the input currency:");
        Scanner in = new Scanner(System.in);
        int choice = in.nextInt();

       String inType = null;
        switch(choice) {
        case 1: inType = "US Dollars >> " + us_dollar_sym;  break;
        case 2: inType = "Euros >> " + euro_sym; break;
        case 3: inType = "British Pounds >> " + pound_sym; break;
        case 4: inType = "Japanese Yen >> " + yen_sym; break;
        default:
        System.out.println("Please restart the program & enter a number from the list.");
        return;
       }

        System.out.println("Please choose the output currency");
        int output = in.nextInt();

        System.out.printf("Now enter the input in " + inType);
        double input = in.nextDouble();

      if (choice == output)
        System.out.println("Same currency no need to convert");
      if (choice == 1 && output == 2)
        {
           double dollar_euro_rate = 0.82;
           rate = input * dollar_euro_rate;
           System.out.printf( "%s" + input + " at a conversion rate of " + dollar_euro_rate + " Dollars to %s
= %.2f\n", (char)us_dollar_sym, euro, rate);
        }
        else if (choice == 1 && output == 3){
           double dollar_pound_rate = 0.70;
           rate = input * dollar_pound_rate;
           System.out.printf( "%s" + input + " at a conversion rate of " + dollar_pound_rate + " Dollars to
%s = %.2f\n", (char)us_dollar_sym, pound, rate);
        }
        else if (choice == 1 && output == 4){
          double dollar_yen_rate = 105.61;
          rate = input * dollar_yen_rate;
          System.out.printf( "%s" + input + " at a conversion rate of " + dollar_yen_rate + " Dollars to %s =
%.2f\n", (char)us_dollar_sym, yen, rate);
        }
      if (choice == 2 && output == 1)
      {
          double euro_dollar_rate = 1.22;
          rate = input * euro_dollar_rate;
          System.out.printf( "%s" + input + " at a conversion rate of " + euro_dollar_rate + " Euros to %s =
%.2f\n", (char)euro_sym, us_dollar, rate);
        }
        else if (choice == 2 && output == 3)
        {
          double euro_pound_rate = 0.86;
          rate = input * euro_pound_rate;
          System.out.printf( "%s" + input + " at a conversion rate of " + euro_pound_rate + " Euros to %s =
%.2f\n", (char)euro_sym, pound, rate);
        }
        else if (choice == 2 && output == 4)
        {
          double euro_yen_rate = 128.54;
          rate = input * euro_yen_rate;
          System.out.printf( "%s" + input + " at a conversion rate of " + euro_yen_rate + " Euros to %s =
%.2f\n", (char)euro_sym, yen, rate);
        }
      if (choice == 3 && output == 1)
      {
          double pound_dollar_rate = 1.42;
          System.out.printf( "%s" + input + " at a conversion rate of " + pound_dollar_rate + " Pounds to
%s = %.2f\n", (char)pound_sym, us_dollar, rate);
        }
        else if (choice == 3 && output == 2)
        {
          double pound_euro_rate = 1.17;
          System.out.printf( "%s" + input + " at a conversion rate of " + pound_euro_rate + " Pounds to %s
= %.2f\n", (char)pound_sym, euro, rate);
        }
        else if (choice == 3 && output == 4)
        {
          double pound_yen_rate = 149.75;
          System.out.printf( "%s" + input + " at a conversion rate of " + pound_yen_rate + " Pounds to %s
= %.2f\n", (char)pound_sym, yen, rate);
        }
      if (choice == 4 && output == 1)
      {
          double yen_dollar_rate = 0.0095;
          System.out.printf( "%s" + input + " at a conversion rate of " + yen_dollar_rate + " Yen to %s =
%.2f\n", (char)yen_sym, us_dollar, rate);
        }
        else if (choice == 4 && output == 2)
        {
          double yen_euro_rate = 0.0078;
          System.out.printf( "%s" + input + " at a conversion rate of " + yen_euro_rate + " Yen to %s =
%.4f\n", (char)yen_sym, euro, rate);
        }
        else if (choice == 4 && output == 3)
        {
          double yen_pound_rate = 0.0067;
          System.out.printf( "%s" + input + " at a conversion rate of " + yen_pound_rate + " Yen to %s =
%.2f\n", (char)yen_sym, pound, rate);
        }
       System.out.println("Thank you for using the currency converter");
    }
      public static void main(String args[])
      {
                currency c=new currency();
      }
}
```

OUTPUT:

```
E:\PJ>javac currency.java

E:\PJ>java currency
Welcome to the Currency Converter Program
Use the following codes to input your currency choices:
 1 - US dollars
 2 - Euros
 3 - British Pounds
 4 - Japanese Yen
Please choose the input currency:
1
Please choose the output currency
3
Now enter the input in US Dollars >> $89
$89.0 at a conversion rate of 0.7 Dollars to Pounds = 62.30
Thank you for using the currency converter
```

INFERENCE:

VIVA QUESTIONS:

1. What is a constructor?
2. What is the difference between a constructor and a method?
3. What is Garbage collection?
4. Compare Garbage collection and finalize method?
5. Why is main method assigned as public?

Ex.No.	**4**	**IMPLEMENTATION OF MULTI-THREADED PROGRAM**
Date		

AIM

To develop a java program for implementing a multi-thread concept.

ALGORITHM

Step 1: Start the program.

Step 2: Create three threads using the class thread.

Step 3: Create classes and methods to calculate the square and cubic value.

Step 4: Calculate the square and cubic value for a random integer using sleep () method.

Step 5: Execute the program.

Step 6: Stop the program.

PROGRAM:

```
import java.util.Random;
class Square extends Thread
{
        int x;
        Square(int n)
        {
                x = n;
        }
        public void run()
        {
                int sqr = x * x;
                System.out.println("Square of " + x + " = " + sqr );
        }
}
class Cube extends Thread
{
        int x;
        Cube(int n)
        {
                x = n;
        }
        public void run()
        {
                int cub = x * x * x;
                System.out.println("Cube of " + x + " = " + cub );
        }
}
class Number extends Thread
{
        public void run()
        {
                Random random = new Random();
                for(int i =0; i<10; i++)
                {
                        int randomInteger = random.nextInt(100);
                        System.out.println("Random Integer generated : " + randomInteger);
                        if(randomInteger%2==0)
                        {
                                Square s = new Square(randomInteger);
                                s.start();
                        }
                        else
                        {
                                Cube c = new Cube(randomInteger);
                                c.start();
```

```
            }
            try
            {
                Thread.sleep(1000);
            }
            catch (InterruptedException ex)
            {
                System.out.println(ex);
            }
        }
    }
}
public class rand
{
    public static void main(String args[])
    {
        Number n = new Number();
        n.start();
     }
}
```

OUTPUT:

```
E:\PJ>javac rand.java
E:\PJ>java rand
Random Integer generated : 12
Square of 12 = 144
Random Integer generated : 31
Cube of 31 = 29791
Random Integer generated : 89
Cube of 89 = 704969
Random Integer generated : 41
Cube of 41 = 68921
Random Integer generated : 24
Square of 24 = 576
Random Integer generated : 71
Cube of 71 = 357911
Random Integer generated : 64
Square of 64 = 4096
Random Integer generated : 12
Square of 12 = 144
Random Integer generated : 14
Square of 14 = 196
Random Integer generated : 95
Cube of 95 = 857375
```

INFERENCE:

VIVA QUESTIONS:

1. What are the high-level thread states?

2. Define multi-threading

3. What is the purpose of the wait (), notify (), and notifyAll () methods?

4. What is the difference between Thread and Process in Java?

5. What are the states in the lifecycle of a Thread?

Ex.No.	**5**	**IMPLEMENTATION OF INTEGER DIVISION USING USER INTERFACE**
Date		

AIM

To develop a java program that creates a user interface to perform integer divisions.

ALGORITHM

Step 1: Start the program

Step 2: Import java.awt, awt. event, applet. Applet packages.

Step 3: Create a Class which implements ActionListener

Step 4: Create Buttons and Text Fields.

Step 5: Get the input from user.

Step 6: Perform the division.

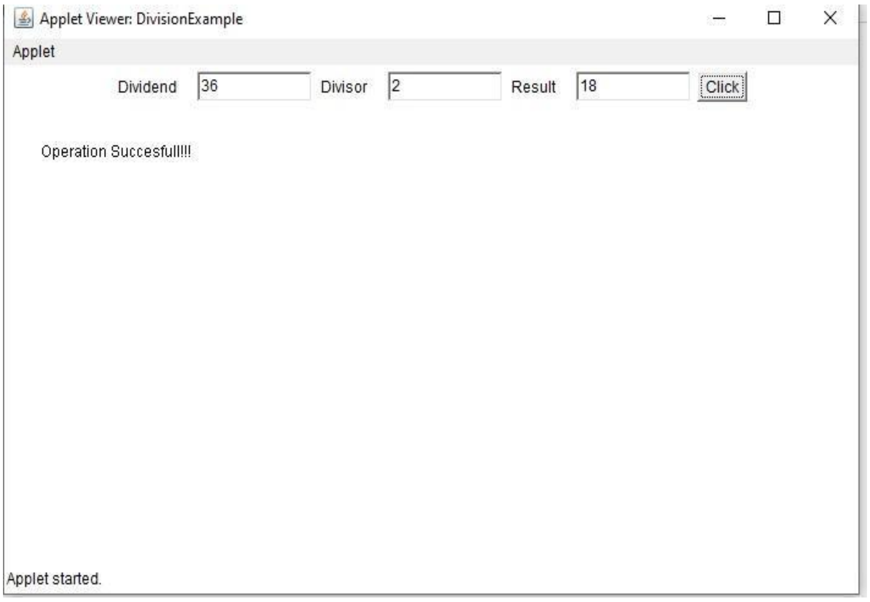
Step 7: Print the Data.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="DivisionExample"width=230 height=250></applet>*/
public class DivisionExample extends Applet implements ActionListener {
        String msg;
        TextField num1, num2, res;
        Label l1, l2, l3;
        Button div;

        public void init() {
                l1 = new Label("Dividend");
                l2 = new Label("Divisor");
                l3 = new Label("Result");
                num1 = new TextField(10);
                num2 = new TextField(10);
                res = new TextField(10);
                div = new Button("Click");
                div.addActionListener(this);
                add(l1);
                add(num1);
                add(l2);
                add(num2);
                add(l3);
                add(res);
                add(div);
        }

        public void actionPerformed(ActionEvent ae) {
                String arg = ae.getActionCommand();
                int num1 = 0, num2 = 0;
                if (arg.equals("Click")) {
                        if (this.num1.getText().isEmpty() && this.num2.getText().isEmpty()) {
                                msg = "Enter the valid numbers!";
                                repaint();
                        } else {
                                try {
                                        num1 = Integer.parseInt(this.num1.getText());
                                        num2 = Integer.parseInt(this.num2.getText());
```

```
                    int num3 = num1 / num2;
                    res.setText(String.valueOf(num3));
                    msg = "Operation Succesfull!!!";
                    repaint();
                } catch (NumberFormatException ex) {
                    System.out.println(ex);
                    res.setText("");
                    msg = "NumberFormatException - Non-numeric";
                    repaint();
                } catch (ArithmeticException e) {
                    System.out.println("Can't be divided by Zero" + e);
                    res.setText("");
                    msg = "Can't be divided by Zero";
                    repaint();
                }
            }
        }
    }
    public void paint(Graphics g) {
        g.drawString(msg, 30, 70);
    }
}
```

OUTPUT:



36

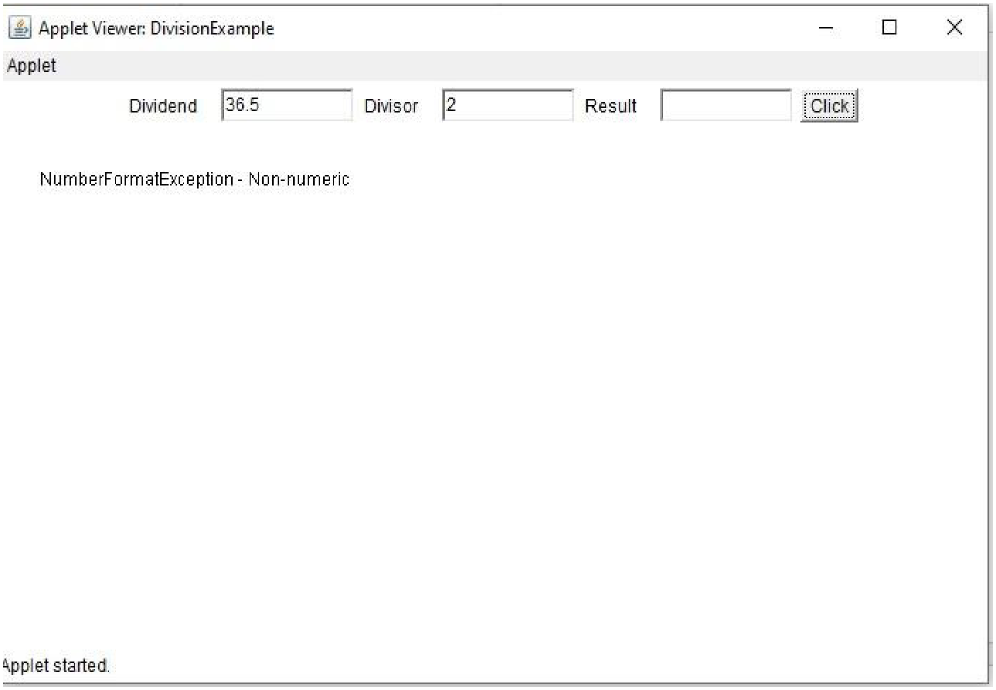
2

18

Operation Succesfull!!!

Applet started.

Numberformat Exception



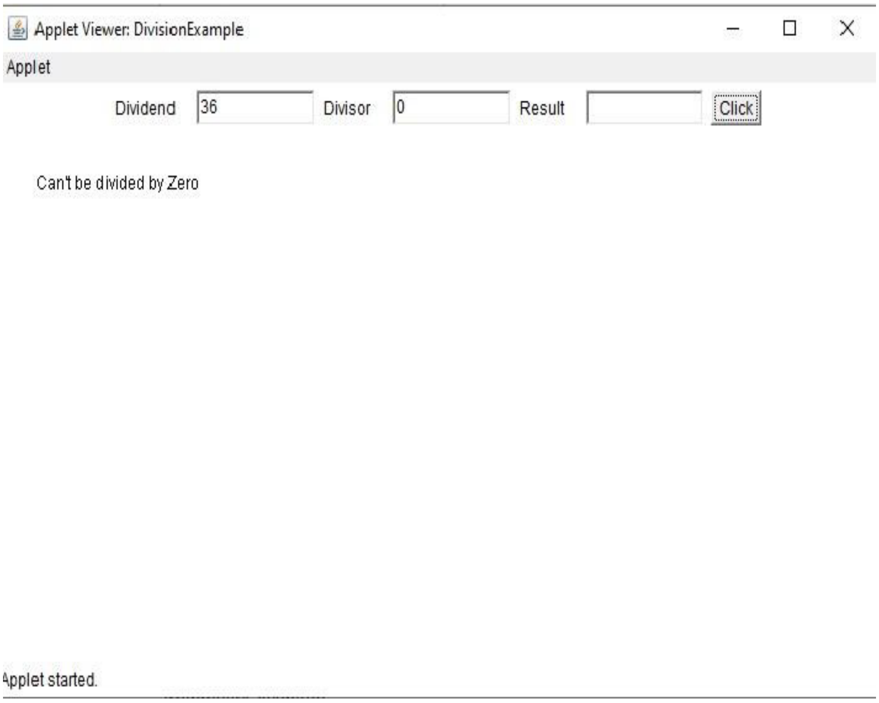
36.5

2

NumberFormatException - Non-numeric

Applet started.

ArithmeticException



Applet Viewer: DivisionExample

Applet

Dividend 36 Divisor 0 Result

Can't be divided by Zero

Applet started.

INFERENCE:

VIVA QUESTIONS:

1. What is an Exception in Java?

2. What are the Exception Handling Keywords in Java?

3. What is the difference between the throw and throws keyword in Java?

4. Define method overriding in java.

5. What is the difference between final, finally, and finalize in Java?

Ex.No.	**6**	**Implementation of common functionalities using interface**
Date		

AIM:

To write a java program using interface with common functionalities of different class.

ALGORITHM:

Step 1: Start the program

Step 2: Create the interface and declare the methods

Step 3: Create a class with a name bicycle, car and bike which implements the interface.

Step 4: Implements all the methods declared in the interface.

Step 5: Stop the program.

PROGRAM

```
interface Vehicle
{
  public Integer getNumberOfSeats();
  public Integer getNumberOfWheels();
  public String getVehicleType();
}
class bicycle implements Vehicle
{
public Integer getNumberOfSeats()
{
 return 1;
}
public Integer getNumberOfWheels()
{
 return 2;
}
public String getVehicleType()
```

```
{
return "BiCycle";
}
}
class Car implements Vehicle
{
public Integer getNumberOfSeats()
{
return 5;
}
public Integer getNumberOfWheels()
{
 return 4;
 }
public String getVehicleType()
{
return "Car";
}
}
class bike implements Vehicle
{
public Integer getNumberOfSeats()
{
       return 2;
}
public Integer getNumberOfWheels()
{
   return 2;
}
public String getVehicleType()
{
   return "Bike";
}
}
public class Interfacemain
{
   public static void main(String args[])
   {
   Vehicle v=new bicycle();
   System.out.println("My "+v.getVehicleType()+" has "+v.getNumberOfSeats()+" Seats and "+v.getNumberOfWheels()+" Wheels");
   Vehicle v1=new Car();
   System.out.println("My "+v1.getVehicleType()+" has "+v1.getNumberOfSeats()+" Seats and "+v1.getNumberOfWheels()+" Wheels");
Vehicle v2=new bike();
System.out.println("My "+v2.getVehicleType()+" has "+v2.getNumberOfSeats()+" Seats and "+v2.getNumberOfWheels()+" Wheels");
   }
}
```

OUTPUT:

```
D:\Jayanthi\java>javac Interfacemain.java

D:\Jayanthi\java>java Interfacemain

My BiCycle has 1 Seats and 2 Wheels

My Car has 5 Seats and 4 Wheels

My Bike has 2 Seats and 2 Wheels
```

INFERENCE

VIVA QUESTIONS:

1. Define an interface.

2. Define abstract methods.

3. Can we re-assign a value to a field of interfaces?

4. Can we override an interface method with visibility other than public?

5. Can we declare an Interface with “abstract” keyword?

RESULT:

Thus the java program has been executed successfully using interface to implement the common functionalities.

Ex.No.	7	IMPLEMENTATION OF BASIC OPERATIONS IN CALCULATOR
Date		

AIM:

To write a java program for implementing basic operations of calculator like addition, subtraction, multiplication and division.

ALGORITHM:

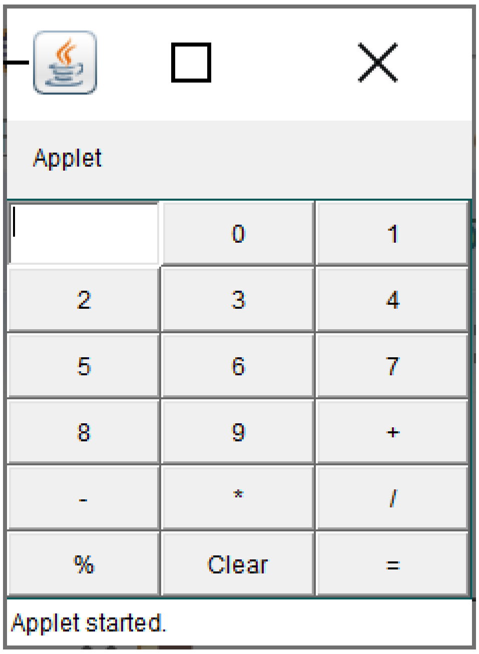
1. Start the program
2. Create a text field to accept the expression and display the output also.
3. Create buttons for digits and a decimal point.
4. Create a button to clear the complete expression.
5. Create the buttons for operations, that is for addition, subtraction, multiplication and division and an equal's button to compute the result.
6. Add Action Listener to all the buttons.
7. Compute the result and display in the text field.
8. Stop the program

PROGRAM

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

/*
  <applet code="Calculator" width=500 height=500></applet>
*/

public class Calculator extends Applet implements ActionListener
{
        String msg=" ";
        int v1,v2,result;
        TextField t1;
        Button b[]=new Button[10];
        Button add,sub,mul,div,clear,mod,EQ;
        char OP;
        public void init()
        {
                t1=new TextField(100);
                GridLayout gl=new GridLayout(6,3);
                setLayout(gl);
                for(int i=0;i<10;i++)
                {
                        b[i]=new Button(""+i);
                }
                add=new Button("+");
                sub=new Button("-");
                mul=new Button("*");
                div=new Button("/");
                mod=new Button("%");
                clear=new Button("Clear");
        EQ=new Button("=");
        t1.addActionListener(this);
        add(t1);
        for(int i=0;i<10;i++)
        {
            add(b[i]);
        }
        add(add);
        add(sub);
        add(mul);
        add(div);
        add(mod);
        add(clear);
        add(EQ);
        for(int i=0;i<10;i++)
        {
            b[i].addActionListener(this);
        }
        add.addActionListener(this);
        sub.addActionListener(this);
        mul.addActionListener(this);
        div.addActionListener(this);
        mod.addActionListener(this);
        clear.addActionListener(this);
        EQ.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae)
    {
        String str=ae.getActionCommand();
        char ch=str.charAt(0);

        if ( Character.isDigit(ch))
            t1.setText(t1.getText()+str);
        else
            if(str.equals("+"))
            {
                v1=Integer.parseInt(t1.getText());
                OP='+';
                t1.setText("");
            }
            else if(str.equals("-"))
            {
                v1=Integer.parseInt(t1.getText());
                OP='-';
                t1.setText("");
            }
            else if(str.equals("*"))
            {
                v1=Integer.parseInt(t1.getText());
                OP='*';
                t1.setText("");
            }
            else if(str.equals("/"))
            {
```

```
				v1=Integer.parseInt(t1.getText());
				OP='/';
				t1.setText("");
			}
			else if(str.equals("%"))
			{
				v1=Integer.parseInt(t1.getText());
				OP='%';
				t1.setText("");
			}

		if(str.equals("="))
		{
			v2=Integer.parseInt(t1.getText());
			if(OP=='+')
				result=v1+v2;
			else if(OP=='-')
				result=v1-v2;
			else if(OP=='*')
				result=v1*v2;
			else if(OP=='/')
				result=v1/v2;
			else if(OP=='%')
				result=v1%v2;
			t1.setText(""+result);
		}
		if(str.equals("Clear"))
		{
			t1.setText("");
		}
	}
}
```

OUTPUT



Applet started.

INFERENCE

VIVA QUESTIONS

1. Define Applet.

2. How will you initialize an applet?

3. How will you initialize an applet?

4. What is grid layout?

5. What is action listener?

RESULT:

Thus the java program has been implemented successfully to perform basic operations of calculator like addition, subtraction, multiplication and division.

Ex.No.	8	IMPLEMENTATION OF APPLET PROGRAM TO HANDLE MOUSE EVENTS
Date		

AIM

To write the java applet program to handle mouse events line entered, clicked, pressed, released or exited.

ALGORITHM

Step 1: Start the program
Step 2: Extends the Applet class and implements the Mouse Listener
Step 3: Include the event for mouse clicked.
Step 4: Include the event for mouse pressed.
Step 5: Include the event for mouse released.
Step 6: Include the event for mouse entered.
Step 7: Include the event for mouse exited
Step 8: Execute the program
Step 9: Stop the program.

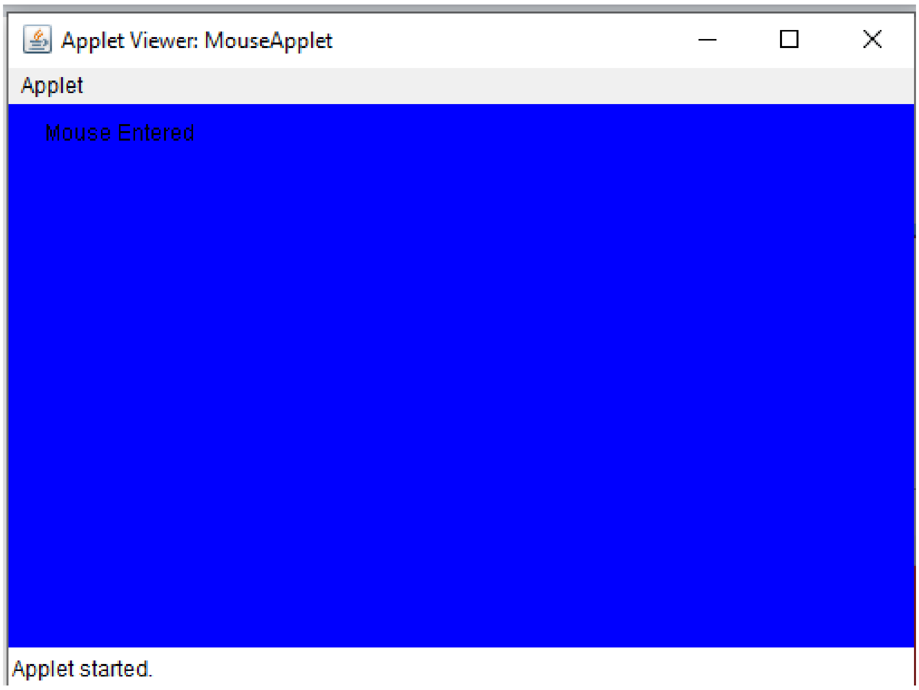
PROGRAM

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.Color;
public class MouseApplet extends Applet implements MouseListener
{
        String msg="Initial Message";
        public void init()
        {
                addMouseListener(this);
        }
        public void mouseClicked(MouseEvent me)
        {
                msg = "Mouse Clicked";
                setBackground(Color.YELLOW);
                repaint();
        }
        public void mousePressed(MouseEvent me)
        {
                msg = "Mouse Pressed";
                setBackground(Color.RED);
                repaint( );
        }
        public void mouseReleased(MouseEvent me)
        {
                msg = "Mouse Released";
                setBackground(Color.GREEN);
                repaint( );
        }
```

```
    public void mouseEntered(MouseEvent me)
    {
        msg = "Mouse Entered";
        setBackground(Color.BLUE);
        repaint();
    }
    public void mouseExited(MouseEvent me)
    {
        msg = "Mouse Exited";
        setBackground(Color.PINK);
        repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString(msg,20,20);
    }
}
/*
<applet code="MouseApplet" height="300" width="500">
</applet>
*/
```

Output:

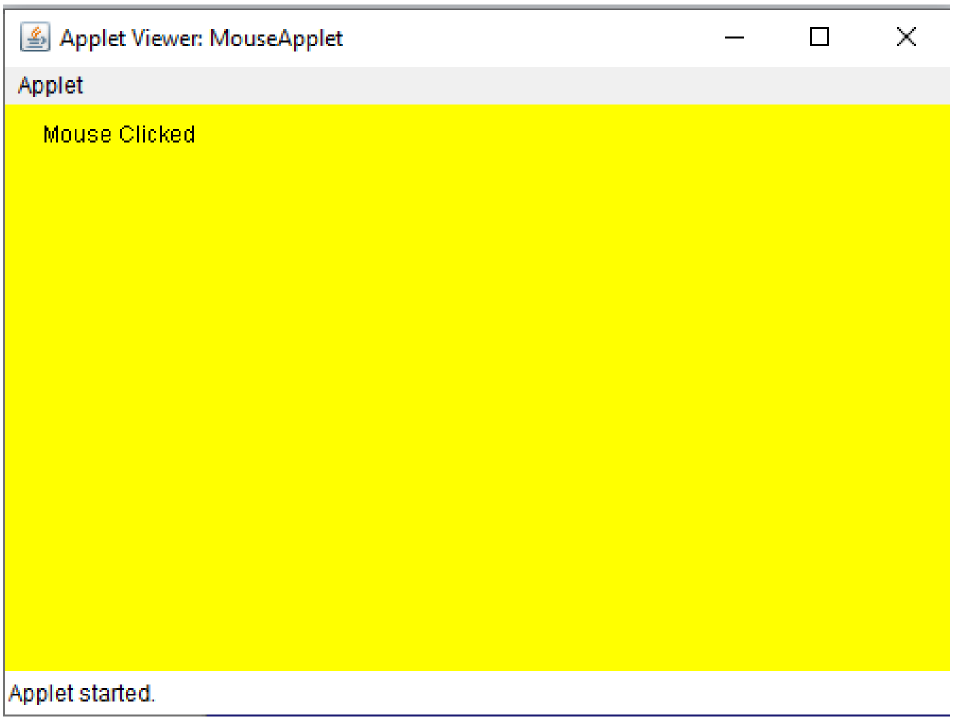
Mouse Entered



Mouse Entered

Applet started.

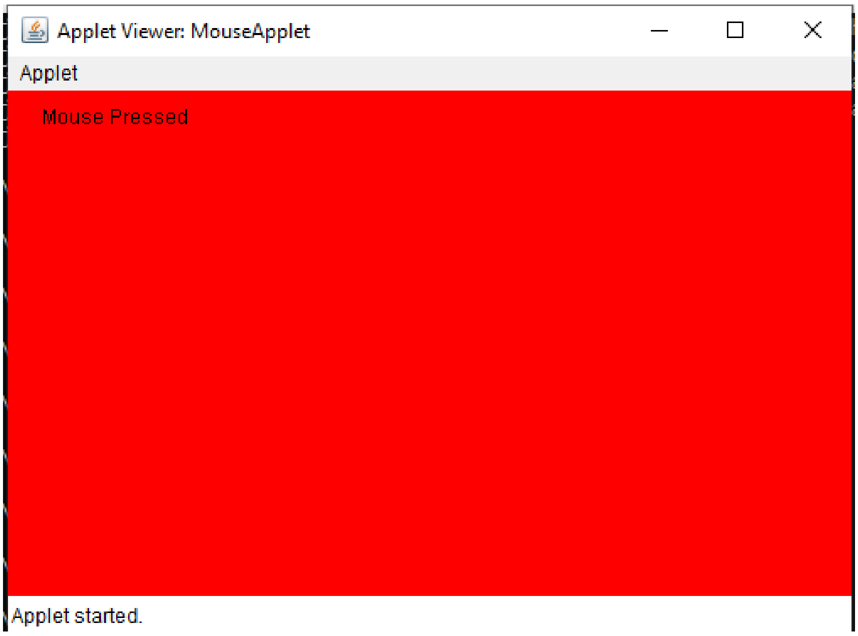
Mouse Clicked



Mouse Clicked

Applet started.

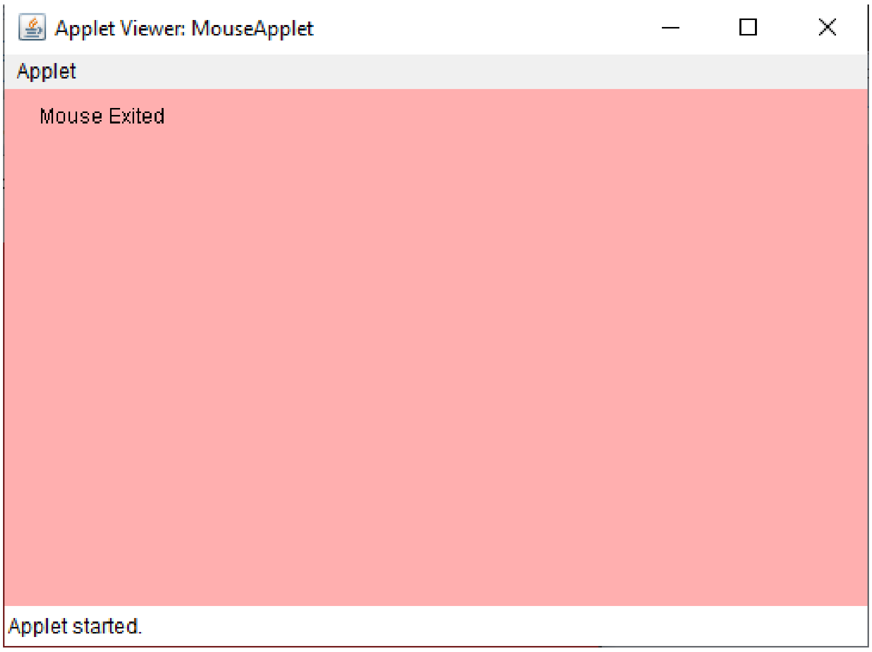
Mouse Pressed



Mouse Pressed

Applet started.

Mouse Exited



Mouse Exited

Applet started.

INFERENCE

VIVA QUESTION

1. What is an adapter class?

2. What interface is extended by AWT event listener?

3. What is an event and what are the models available for event handling?

4. What are the advantages of the model over the event inheritance model?

5. What is action listener?

RESULT

Thus the java applet program to handle mouse events has been written and executed successfully.

Ex.No.	9	Implementation Of Area Of Different Shapes Using Abstract Class
Date		

AIM:

To write a java program to find the area of Rectangle, Triangle and Circle using an Abstract class.

ALGORITHM:

```
Step 1: Start the program
Step 2: Create an abstract class with a method print_area()
Step 3: Create a class with a name Rectangle, Triangle and Circle which inherits the abstract class.
Step 4: Calculate the area of Rectangle, Triangle and Circle using the abstract method print_area().
Step 5: Execute the program and find the result.
Step 6: Stop the program.
```

PROGRAM:

```
import java.util.*;
abstract class shape
{
int x,y;
abstract void print_area(double x,double y);
}
class Rectangle extends shape
{
void print_area (double x,double y)
{
System.out.println("Area of rectangle is :"+(x*y));
}
}
class Circle extends shape
{
void print_area (double x,double y)
{
System.out.println("Area of circle is :"+(3.14*x*x));
}
}
class Triangle extends shape
{
void print_area(double x,double y)
{
System.out.println("Area of triangle is :"+(0.5*x*y));
}
}
public class AbstactDemo
{
public static void main(String[] args)
{
Rectangle r=new Rectangle();
r. print_area(2,5);
Circle c=new Circle();
c. print_area(5,5);
Triangle t=new Triangle();
t. print_area (2,5);
}
}
```

OUTPUT:

```
Area of rectangle is :10.0
Area of circle is :78.5
Area of triangle is :5.0
```

INFERENCE:

In this program I learnt about the abstract class.

VIVA QUESTIONS

1. Define Abstract class in java

2. Can abstract class implements interface in Java? Do they require to implement all methods?

3. Is it necessary for an abstract class to have an abstract method?

4. When do you favor abstract class over interface?

5. What is abstract method in Java?

RESULT:

Thus a java program to find the area of Rectangle, Triangle and Circle using an Abstract class has been executed successfully.

Ex.No.	10	Simulation of traffic light using an applet.
Date		

AIM:

To write a java program to simulate a traffic light using an applet.

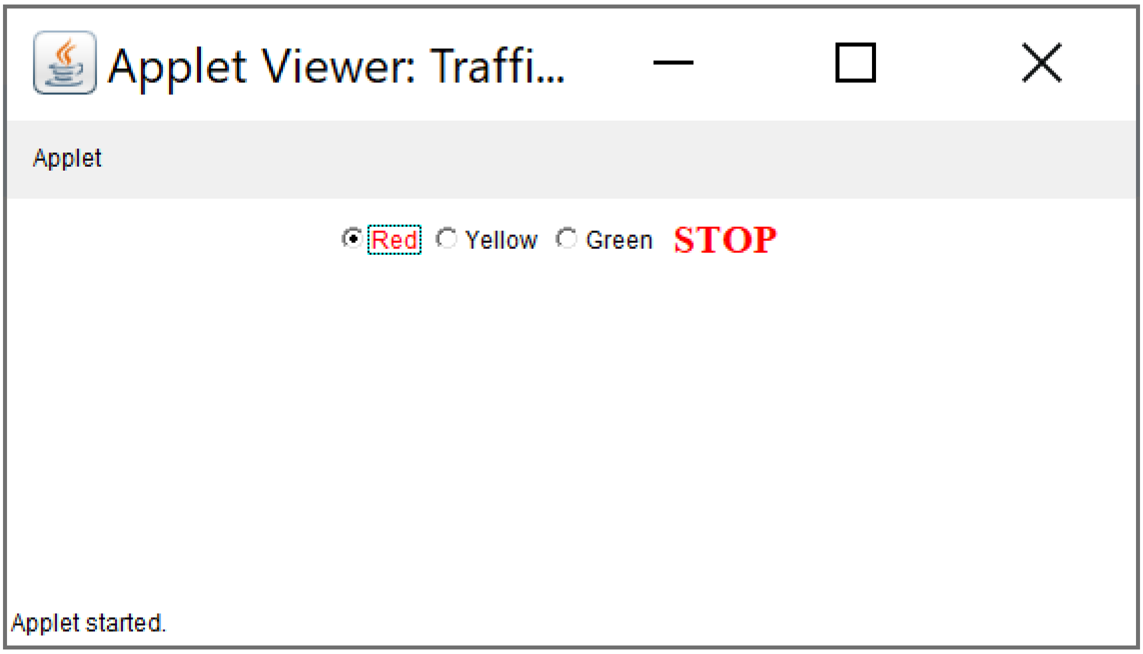
ALGORITHM:

Step 1: Start the program
Step 2: Create an Applet.
Step 3: Create the button for the lights
Step 4: Add Listener to the button.
Step 5: Include the text message to be displayed while choosing the colors.
Step 6: Execute the program
Step 7: Stop the program.

PROGRAM

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

/*
 * <applet code = "TrafficLightsExample" width = 1000 height = 500>
 * </applet>
 * */

public class TrafficLightsExample extends Applet implements ItemListener{

        CheckboxGroup grp = new CheckboxGroup();
        Checkbox redLight, yellowLight, greenLight;
        Label msg;
        public void init(){
                redLight = new Checkbox("Red", grp, false);
                yellowLight = new Checkbox("Yellow", grp, false);
                greenLight = new Checkbox("Green", grp, false);
                msg = new Label("");

                redLight.addItemListener(this);
                yellowLight.addItemListener(this);
                greenLight.addItemListener(this);

                add(redLight);
                add(yellowLight);
                add(greenLight);
                add(msg);
                msg.setFont(new Font("Serif", Font.BOLD, 20));
        }
        public void itemStateChanged(ItemEvent ie) {
                redLight.setForeground(Color.BLACK);
                yellowLight.setForeground(Color.BLACK);
                greenLight.setForeground(Color.BLACK);


                if(redLight.getState() == true) {
                        redLight.setForeground(Color.RED);
                        msg.setForeground(Color.RED);
                        msg.setText("STOP");
                }
                else if(yellowLight.getState() == true) {
                        yellowLight.setForeground(Color.YELLOW);
                        msg.setForeground(Color.YELLOW);
                        msg.setText("READY");
                }
                else{
                        greenLight.setForeground(Color.GREEN);
                        msg.setForeground(Color.GREEN);
                        msg.setText("GO");
                }
        }
}
```

OUTPUT:



STOP

Applet started.

INFERENCE

By executing this program I have learnt to use the buttons, listener and to display the message while choosing the colors.

VIVA QUESTIONS

1. What is Item Listener?
2. How to group the checkbox?
3. How to display the message box?
4. How to add listener to the check box?
5. What is item state change?

RESULT:

Thus a java program has been written to simulate a traffic light using an applet and executed successfully.

Ex.No.	**11**	**Implementation Of Area Of Different Shapes Using Method Overloading**
Date		

AIM

To write a java program that displays area of different Figures using the method overloading.

ALGORITHM

Step 1: Start the program.
Step 2: Overload the area method to find the area of square which takes one float parameter.
Step 3: Overload the area method to find the area of rectangle which takes two float parameters.
Step 4: Overload the area method to find the area of circle which takes one double parameter.
Step 5: Create an object for OverloadDemo class and call the area() method.
Step 6: Execute the program
Step 7: Stop the program.

PROGRAM

```
class OverloadDemo
{
   void area(float x)
   {
      System.out.println("The area of the square is "+Math.pow(x, 2)+" sq units");
   }
   void area(float x, float y)
   {
      System.out.println("The area of the rectangle is "+x*y+" sq units");
   }
   void area(double x)
   {
      double z = 3.14 * x * x;
      System.out.println("The area of the circle is "+z+" sq units");
   }
}
class Overload
{
    public static void main(String args[])
         {
           OverloadDemo ob = new OverloadDemo();
           ob.area(5);
           ob.area(11,12);
           ob.area(2.5);
      }
}
```

OUTPUT

```
D:\pj>javac Overload.java

D:\pj>java Overload
The area of the square is 25.0 sq units
The area of the rectangle is 132.0 sq units
The area of the circle is 19.625 sq units
```

\

INFERENCE:

In this program I learn about method overloading concept.

VIVA QUESTIONS:

1. Define method overloading.
2. What is method signature? What are the things it consist of?
3. How do compiler differentiate overloaded methods from duplicate methods?
4. Can we declare one overloaded method as static and another one as non-static?
5. Is it possible to have two methods in a class with same method signature but different return types?

RESULT

Thus a java program have been written to display area of different Figures using the method overloading and executed successfully.

Ex.No.	12	Implementation Of Finding Student Eligibility Using An Interface
Date		

AIM

To write a java program for finding the details of the students eligible to enroll in the examination using interface.

ALGORITHM

```
Step 1: Start the program
Step 2: Create an interface and declare a method print_details().
Step 3: Create a class which inherits the interface to collect the student details.
Step 4: Define print_details() method with eligibility criteria for the students to enroll in the examinations.
Step 5: Execute the program.
Step 6: Check the eligibility criteria and print the result.
Step 7: Stop the program  .
```

PROGRAM

```
import java.util.Scanner;
interface Student
{
        void print_details();
}
class GetStudentDetails implements Student
{

      String name,dept;
       int roll;
        float att;
                Scanner SC=new Scanner(System.in);
                public void get_details()
                {
                        System.out.print("Enter Name: ");
                        name=SC.nextLine();
                        System.out.print("Enter Department: ");
                        dept=SC.nextLine();
                              System.out.print("Enter Roll Number: ");
                              roll=SC.nextInt();
                              System.out.print("Enter Your Attendance percentage: ");
                              att=SC.nextFloat();

                }
                public void print_details()
                {

                              System.out.println("Name: "+name);
                                System.out.println("Department: "+dept);



                                System.out.println("Roll No: "+roll);
                System.out.println("Attendance percentage: "+att);
               if(att>75)
                {
                    System.out.println("Your are eligible to attend the exam");


                }
                else


{
                System.out.println("Your are not eligible to attend the exam");
                }
            }

}
public class eligible
{
    public static void main(String args[])
    {
        GetStudentDetails s=new GetStudentDetails();
        s.get_details();
        s.print_details();
    }
}
```

OUTPUT:

```
D:\pj>java eligible
Enter Name: Ramu
Enter Department: CSE
Enter Roll Number: 78
Enter Your Attendance percentage: 76.4

Name: Ramu
Department: CSE
Roll No: 78
Attendance percentage: 76.4
Your are eligible to attend the exam
```

INFERENCE

In this program I learnt about the interface and its usage.

VIVA QUESTIONS:

1. Define an interface.

2. What will happen if we define a concrete method in an interface in Java?

3. What will happen if we do not initialize variables in Java interface?

4. Can we declare the interface as final?

5. Can an interface extends another interface?

RESULT:

Thus a java program for finding the details of the students eligible to enroll in the examination using interface is written and executed successfully.

Ex.No.	**13**	

Date		**Implementation Of Http Session Tracking Using Servlet**

AIM

To write a java program to use the HttpSession object to find out the creation time and the last-accessed time for a session.

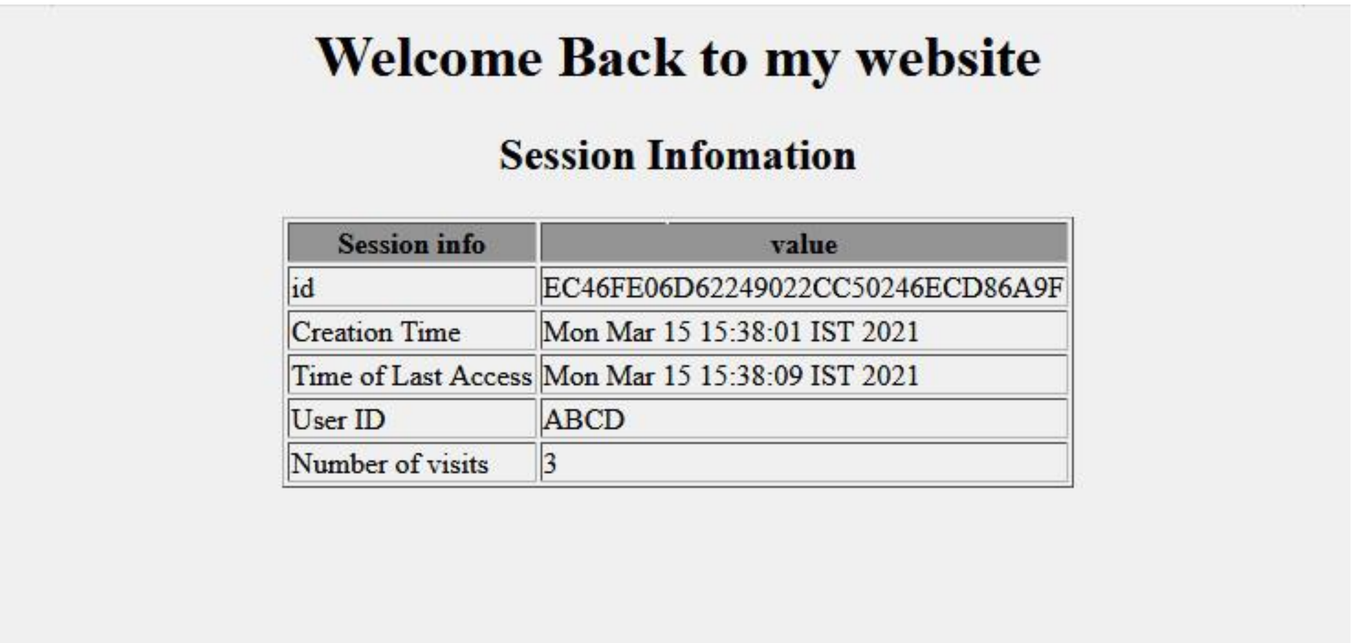
ALGORITHM

```
1.Start the program.
2. Import all the required java libraries.
3.Extends HttpServlet class
4.Create an HttpSession object.
5.Get the session creation time.
6.Get the last access time of this web page
7.Set the response content type.
8.Execute the program
9.Stop the program
```

PROGRAM

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
public class SessionTrack extends HttpServlet {
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
     response.setContentType("text/html;charset=UTF-8");
     try (PrintWriter out = response.getWriter()) {
HttpSession session = request.getSession(true);

    // Get session creation time.
    Date createTime = new Date(session.getCreationTime());

    // Get last access time of this web page.
    Date lastAccessTime = new Date(session.getLastAccessedTime());

    String title = "Welcome Back to my website";
    Integer visitCount = new Integer(0);
    String visitCountKey = new String("visitCount");
    String userIDKey = new String("userID");
    String userID = new String("ABCD");




    // Check if this is new comer on your web page.
    if (session.isNew()) {
      title = "Welcome to my website";
      session.setAttribute(userIDKey, userID);
    } else {
      visitCount = (Integer)session.getAttribute(visitCountKey);
      visitCount = visitCount + 1;
      userID = (String)session.getAttribute(userIDKey);
    }
    session.setAttribute(visitCountKey,  visitCount);

    // Set response content type
    response.setContentType("text/html");
    //PrintWriter out = response.getWriter();

    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" +
          "<h2 align = \"center\">Session Infomation</h2>\n" +
          "<table border = \"1\" align = \"center\">\n" +

            "<tr bgcolor = \"#949494\">\n" +
              "  <th>Session info</th><th>value</th></tr>\n" +

            "<tr>\n" +
              "  <td>id</td>\n" +"  <td>" + session.getId() + "</td></tr>\n" +

            "<tr>\n" +
              "  <td>Creation Time</td>\n" +
              "  <td>" + createTime + "  </td></tr>\n" +

            "<tr>\n" +
              "  <td>Time of Last Access</td>\n" +
              "  <td>" + lastAccessTime + "  </td>  </tr>\n" +

            "<tr>\n" +
              "  <td>User ID</td>\n" +
              "  <td>" + userID + "  </td></tr>\n" +

            "<tr>\n" +
              "  <td>Number of visits</td>\n" +
              "  <td>" + visitCount + "</td></tr>\n" +
          "</table>\n" + "</body></html>"
    );
      }
  }
```

OUTPUT



Welcome Back to my website

Session Infomation

Session info	value
id	EC46FE06D62249022CC50246ECD86A9F
Creation Time	Mon Mar 15 15:38:01 IST 2021
Time of Last Access	Mon Mar 15 15:38:09 IST 2021
User ID	ABCD
Number of visits	3

INFERENCE

In this program I understand how to create the HttpSession using servlet and implement the same using net beans software.

VIVA QUESTIONS:

1. Define Servlet
2. What is the life-cycle of a servlet?
3. Who is responsible to create the object of servlet?
4. What is difference between Print Writer and ServletOutputStream?
5. What is get method and post method?

RESULT

Thus a java program to use the HttpSession object to find out the creation time and the last-accessed time for a session has been written and executed successfully.

Ex.No.	**14**	**Implementation Of Factorial of a number using JSP**
Date		

AIM

To write a JSP program to find the factorial of a number.

ALGORITHM

Step 1:Start the program
Step 2: Create HTML page for giving input
Step 3: Calculate the Factorial using JSP
Step 4: Execute the program
Step 5: Stop the program.

PROGRAM

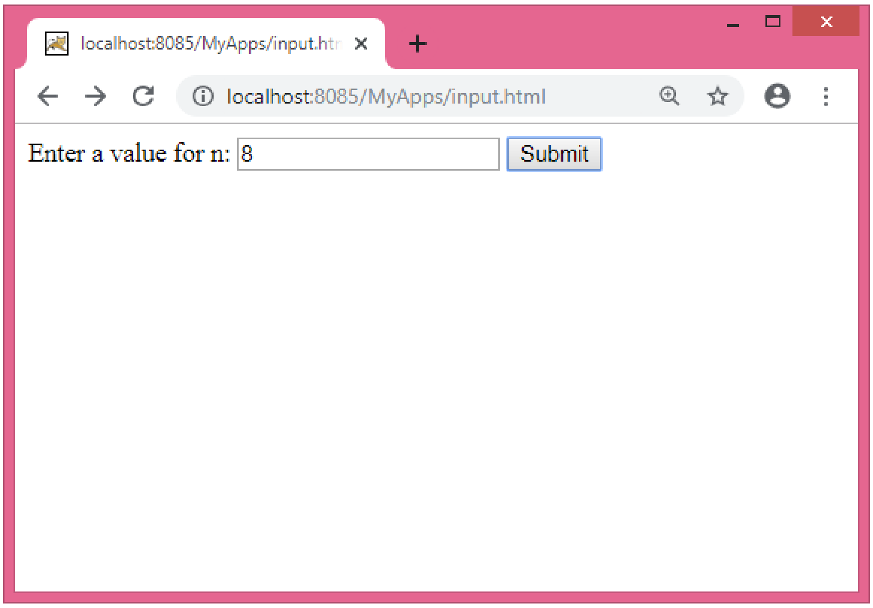
input.html

```
input.html
<html>
<body>
<form action="Factorial.jsp">
Enter a value for n: <input type="text" name="val">
<input type="submit" value="Submit">
</form>
</body>
</html>
```

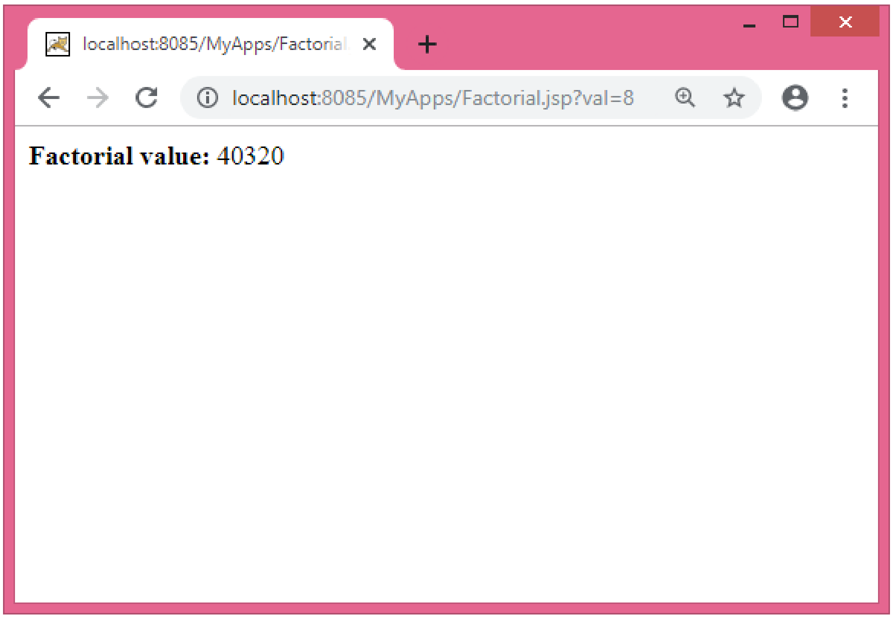
Factorial.jsp

```
<html>
<body>
<%!
  long n, result;
  String str;

  long fact(long n) {
    if(n==0)
      return 1;
    else
      return n*fact(n-1);
  }
%>
<%
   str = request.getParameter("val");
   n = Long.parseLong(str);
   result = fact(n);
%>
<b>Factorial value: </b> <%= result %>
</body>
</html>
```

Output:


localhost:8085/MyApps/input.htr
localhost:8085/MyApps/input.html
Submit

Enter a value for n: 8


localhost:8085/MyApps/Factorial
localhost:8085/MyApps/Factorial.jsp?val=8

Factorial value: 40320

INFERENCE

RESULT

Thus the JSP program to find the factorial of a number has been implemented successfully.