MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

KALITHEERTHALKUPPAM, PUDUCHERRY- 605 107

DEPARTMENT
OF
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

LAB MANUAL
REGULATION (2023-2024)

Staff Name	**:**	**Ms.Renga Nayagi M.E.,**
Designation	**:**	**Assistant Professor / AIML**
Subject Code / Name	**:**	**AMPL302 / Object Oriented Programming Lab**
Regulation	**:**	**2023-2024**
Year / Semester/Sec	**:**	**II / III/A**

SYLLABUS

PRACTICAL EXERCISES:

1. **Program to implement classes and objects.**
2. **Program to implement constructors and destructors with array of objects.**
3. **Program to demonstrate function overloading.**
4. **Program to implement different types of inheritances like multiple, Multilevel and hybrid.**
5. **I/O Program to demonstrate the use of abstract classes.**
6. **Program to demonstrate I/O streams and functions.**
7. **Program to perform all possible type conversions.**
8. **Program to demonstrate Exception Handling Technique.**
9. **Program to implement networking concepts.**
10. **Program to implement RMI concepts.**
11. **Program to implement AWT concepts.**
12. **Program to implement swing concepts.**
13. **Program to design and implement applet.**
14. **Program to design and implement JDBC**
15. **Program to design an event handling event for simulating a simple calculator.**

ADDITIONAL

16. **Develop a java application to implement currency converter (Dollar to INR, EURO to INR, Yen to INR and vice versa), distance converter (meter to KM, miles to KM and vice versa), time converter (hours to minutes, seconds and vice versa) using packages**

Total : 45 PERIODS

TABLE OF CONTENTS

Ex. No: 1	CLASSES AND OBJECTS	Date:

AIM:

To write a C++ program to generate Employee Pay Slip using Class & Objects

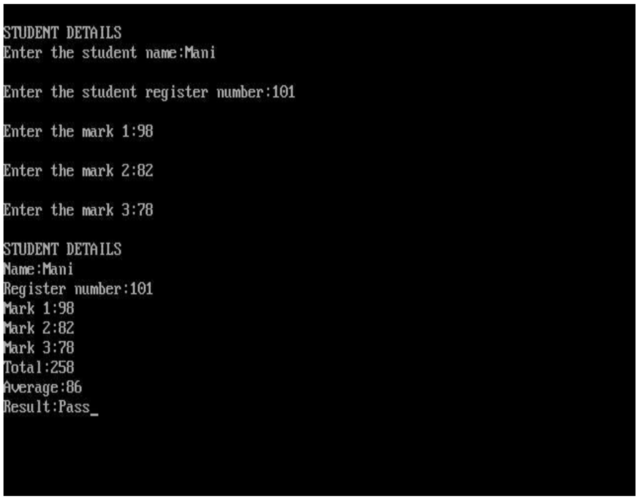
ALGORITHM:

1. Create a class for the Employee with its data member and member function
2. Get number of employee’s n from the user
3. Create object e for the class employee
4. Form a loop for n employees
5. Get the details of the employees
6. Calculate gross pay and net pay
7. Display the employee details
8. Repeat the step 5 until the condition i>n fails
9. Stop the process

CODING:

```
#include<iostream.h>
#include<conio.h>

Class employee                                    //Class Declaration
{
private:
        chare name[20],designation[15];    //Data Member Declaration
        float basic,hra,da,ta,pf,gp,net;

public:
        void get details()                        //Function to get employee details
        {
                cout<<"\n\tEnter the Employee name\t\t\t:";
                cin>>e name;
                cout<<"\n\tEnter the Designation of the Employee :";
                cin>>designation;
                cout<<"\n\tEnter the Basic pay\t\t\t:";
                cin>>basic;
        }

        Void calculation()                        //Function to calculate salary
        {
                da= 0.4*basic;
                hra=0.25*basic;
                pf=0.1*basic;
                ta=0.5*basic;
                gp=basic+da+hra+ta;
                net=gp-pf;
        }

        void display()                          //Function to display employee details
        {
                cout<<"\n\t\t\t EMPLOYEE DETAILS";
                cout<<"\n-----------------------------------------------------";
                cout<<"\n\t Employee name\t:"<<e name;
                cout<<"\n\t Employee Desig :"<<designation;
                cout<<"\n\t Basic pay\t:"<<basic;
                cout<<"\n\t DA\t\t:"<<da;
                cout<<"\n\tHRA\t\t:"<<hra;
                cout<<"\n\tTA\t\t:"<<ta;
                cout<<"\n\tPF\t\t:"<<pf;
                cout<<"\n\tGross pay\t:"<<gp;
                cout<<"\n\tNetpay\t\t:"<<net;
                cout<<"\n-----------------------------------------------------";
         }
};

void main()
{
        clrscr();
        employee e;    //Object declaration

        e.get details();
        e.calculation();
        e.display();

        getch();

}
```

INPUT & OUTPUT:



```
STUDENT DETAILS
Enter the student name:Mani

Enter the student register number:101

Enter the mark 1:98

Enter the mark 2:82

Enter the mark 3:78

STUDENT DETAILS
Name:Mani
Register number:101
Mark 1:98
Mark 2:82
Mark 3:78
Total:258
Average:86
Result:Pass_
```

RESULT:

Thus the program class and objects executed and verified successfully

Ex. No: 2	**CONSTRUCTOR AND DESTRUCTOR**	**Date:**

AIM:

To write a C++ program to demonstrate Constructor and Destructor.

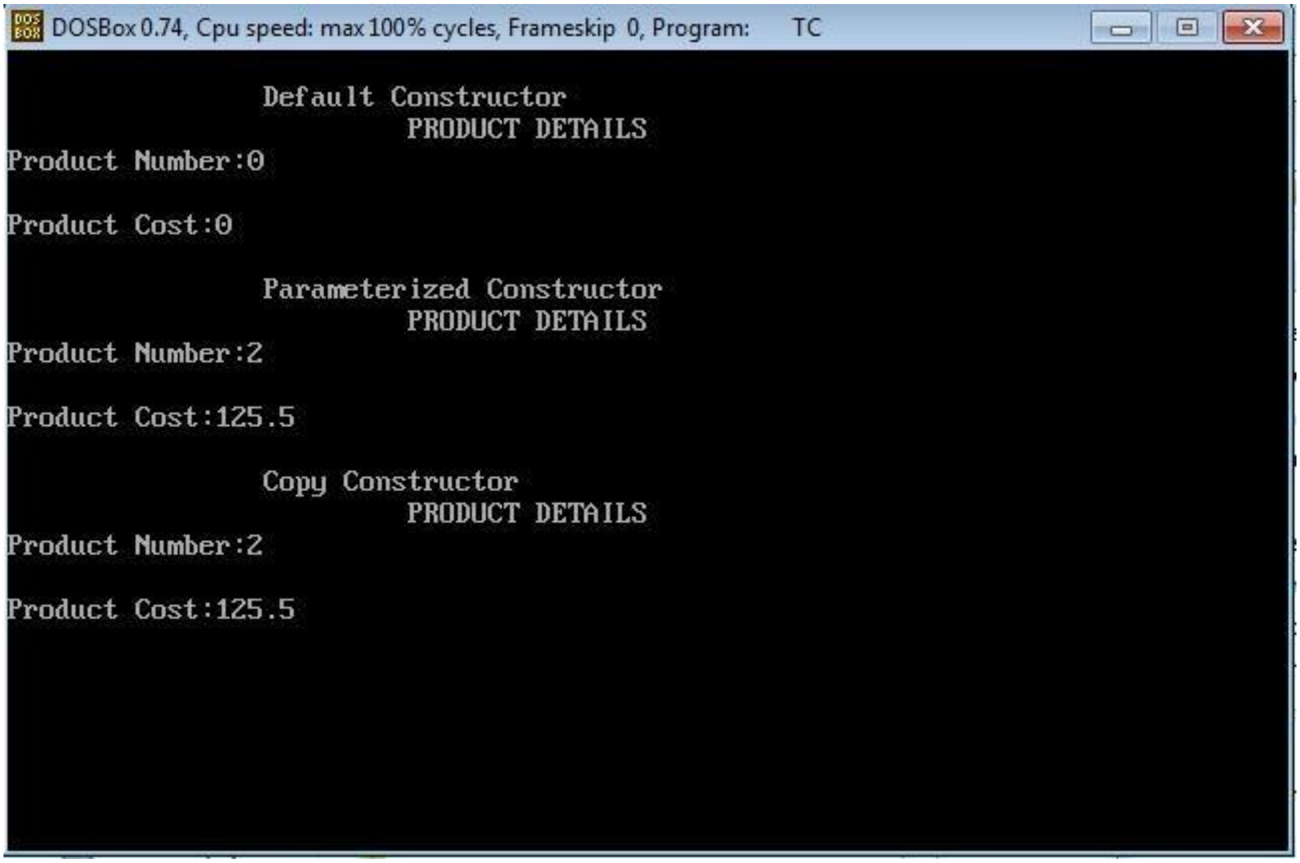
ALGORITHM:

1. Start the program.
2. Declare the class name as product with data members and member functions.
3. The default constructor product() is assigned with the product number as 0 and the cost as 0.0
4. The parameterized constructor product() with argument is assigned with the product number as a and the cost as b.
5. Copy constructor with argument to assign the value.
6. Destructor is used to destroy the product.
7. Display function for displaying the product details.
8. In the main function, p1 object invoke default constructor.
9. p2 object invoke parameterized constructor.
10. P3 object invoke copy constructor.
11. Stop the program

CODING:

```
#include<iostream.h>
#include<conio.h>

Class product                    //Class Declaration
{
        private:                 //Data Member Declaration
        int product no;
        float cost;

        public:
        product()       //Default Constructor to initialize data member
        {
     productno=0; cost=0.0;
        }

        product(int a,float b)//Parameterized Constructor to initialize data member
        {
                productno=a;
                cost=b;
        }
        product(product &x)//Copy Constructor to initialize data member
        {
                productno=x.productno;
                cost=x.cost;
        }

        product()       //Destructor
        {
                cout<<"\nProduct "<<product no<<"is destroyed\n";
        }

        void display(void)      //Function to display product details
        {
                cout<<"\n\t\t\t PRODUCT DETAILS";
                cout<<"\nProduct Number:"<<productno<<"\n";
                cout<<"\nProduct Cost:"<<cost<<"\n";
        }

};

int main()
{
          clrscr();

          productp1;              //Object declaration & Default constructor invoked
          cout<<"\n\t\tDefault Constructor";
          p1.display();

          cout<<"\n\t\tParameterized Constructor ";
          product p2(2,125.5);//Object declaration & Parameterized constructor invoked
          p2.display();

          cout<<"\n\t\t Copy Constructor ";
          product p3(p2);                 //Object declaration & Copy constructor invoked
          p3.display();

          getch();
          return 0;
}
```

INPUT & OUTPUT:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
                Default Constructor
                         PRODUCT DETAILS
Product Number:0

Product Cost:0

                Parameterized Constructor
                         PRODUCT DETAILS
Product Number:2

Product Cost:125.5

                Copy Constructor
                         PRODUCT DETAILS
Product Number:2

Product Cost:125.5
```

RESULT:

Thus the constructors and destructors program executed and verified successfully

Ex. No: 3	**FUNCTION OVERLOADING**	**Date:**

AIM:

To write a C++ program to demonstrate Function Overloading

ALGORITHM:

1. Start the program.
2. Declare the class name as volume with data members and member functions.
3. Read the choice from the user.
4. Choice=1 then go to the step5.
5. The function calculate() to find the side of the cube with one integer argument.
6. Choice=2 then go to the step7.
7. The function calculate() to find the radius and height of the cylinder with two integer argument.
8. Choice=3 then go to the step9.
9. The function calculate () to find the base and height with two integer argument.
10. Choice=4 then stop the program

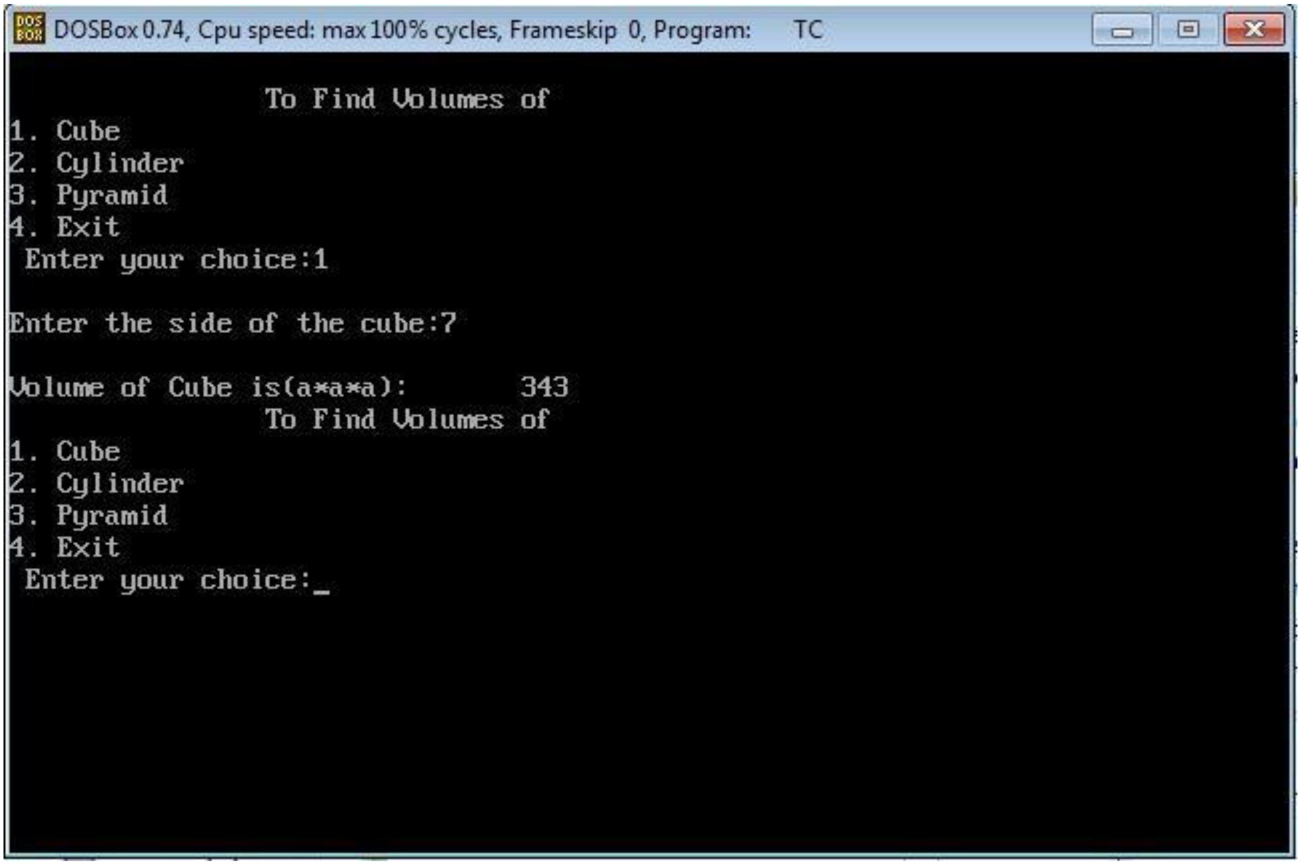
CODING:

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
#include<stdlib.h>

class volume //Class Declaration
{
        public:

          void calculate(int a)          //Overloaded Function to find volume of cube
               {
                       cout<<"\nVolume of Cube is(a*a*a):\t"<<pow(a,3);
               }

         void calculate(floatr,inth)             //Overloaded Function to find volume of cylinder
               {
                       cout<<"\nVolume of cylinder is(3.14*r*r*h):\t"<<3.14*r*r*h;
               }

         void calculate(int b,int h) //Overloaded Function to find volume of pyramid
               {
                       cout<<"\nVolume of pyramid is((1/3)*b*h):\t"<<0.33*b*h;
               }
```

```
};

int main()
{
        clrscr();
       volumev;         //Object declaration

        inta,b,h,ch;     //Variable Declaraion
        floatr;

        do
        {
                cout<<"\n\t\tTo Find Volumes of";
                cout<<"\n1. Cube";
                cout<<"\n2. Cylinder";
                cout<<"\n3. Pyramid";
                cout<<"\n4. Exit";
                cout<<"\n Enter your choice:";
                cin>>ch;

                switch(ch)
                {
                        case 1:
                                cout<<"\nEnter the side of the cube:";
                                cin>>a;
                                v.calculate(a);
                                break;

                        case 2:
                                cout<<"\nEnter the radius of the cylinder:";
                                cin>>r;
                                cout<<"\nEnter the height of the cylinder:";
                                cin>>h;
                                v.calculate(r,h);
                                break;

                         case3:
                                cout<<"\nEnter the base of the pyramid:";
                                cin>>b;
                                cout<<"\nEnter the height of the pyramid:";
                                cin>>h;
                                v.calculate(b,h);
                                break;

                         case4:
                                exit(0);
                                break;
```

```
                    default:
                    cout<<"\nEnter the correct choice";
                }
        }while(ch<=3);

        getch();
        return 0;
}
```

INPUT & OUTPUT:



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
                To Find Volumes of
1. Cube
2. Cylinder
3. Pyramid
4. Exit
 Enter your choice:1

Enter the side of the cube:7

Volume of Cube is(a*a*a):       343
                To Find Volumes of
1. Cube
2. Cylinder
3. Pyramid
4. Exit
 Enter your choice:_
```

RESULT:

Thus the function overloading program executed and verified successfully

Ex. No: 4(A)	**MULTIPLE INHERITANCE**	**Date:**

AIM:

To write a C++ program to implement multiple inheritance using Student information system

ALGORITHM:

1. Start the program.
2. Declare the base class student.
3. Declare and define the function get data() to get the student details.
4. Declare the other class marks.
5. Declare and define the function get marks() to get the student marks.
6. Create the class result derived from student and marks.
7. Declare and define the function calculation () to find out the total and average of the student.
8. Declare the function display() to display the student details.
9. Declare the derived class object and call the functions get data(), get marks(), get calculation() and display().
10. Stop the program

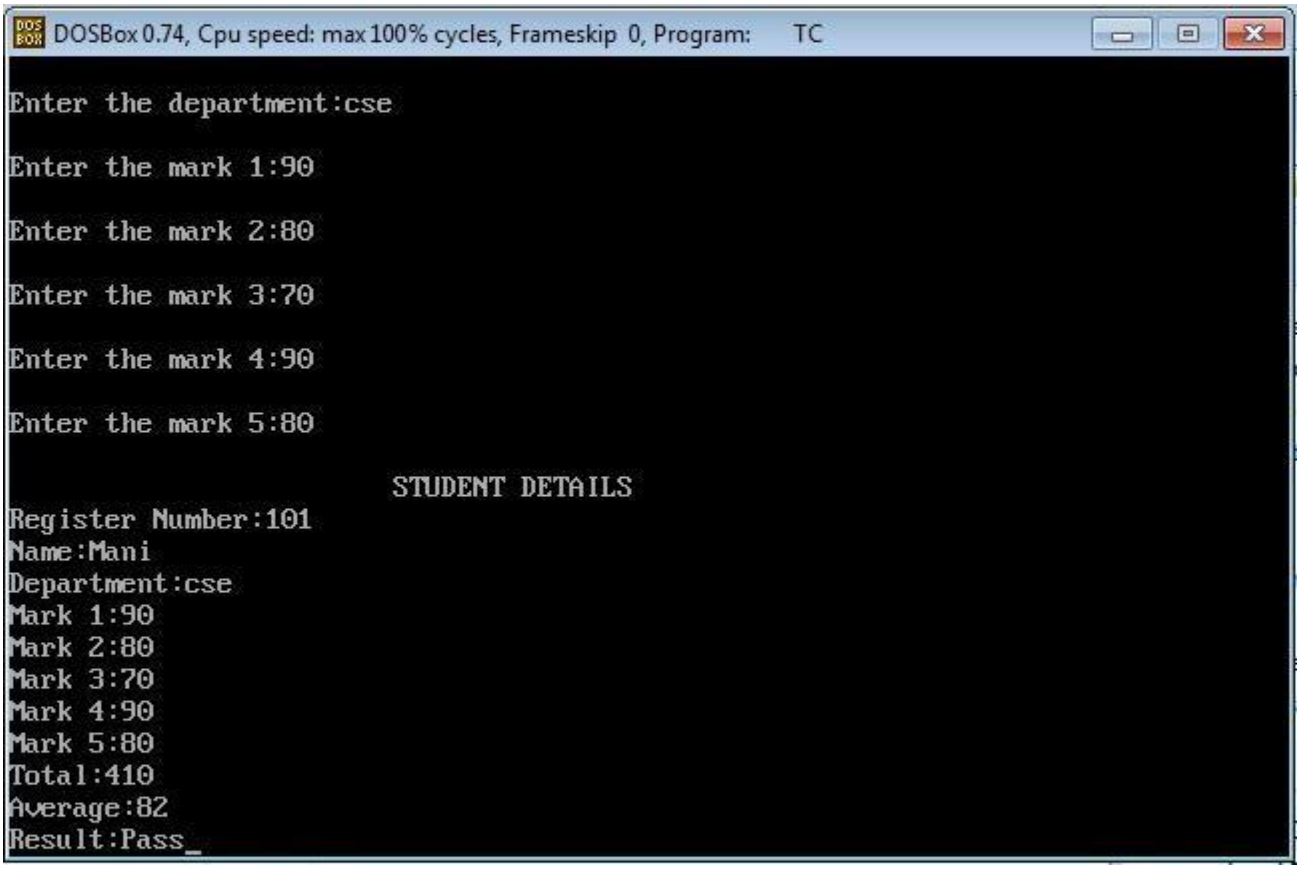
CODING:

```
#include<iostream.h>
#include<conio.h>
#include<string.h>

Class student    //Class Declaration
{
        protected:
                int reg no;                    //Data Member Declaration
                char name[20],dept[5];

        public:
                void get data()                        //Function to get student details
                {
                        cout<<"\nEnter the register number:";
                        cin>>reg no;
                        cout<<"\nEnter the name:";
                        cin>>name;
                        cout<<"\nEnter the department:";
                        cin>>dept;
                }
      };
```

```
 Class marks                    //Class Declaration
{
   protected:
                intm1,m2,m3,m4,m5;                  //Data Member Declaration

   public:
                void get marks()              //Function to get student marks
                {
                        cout<<"\nEnter the mark 1:";
                        cin>>m1;
                        cout<<"\nEnter the mark 2:";
                        cin>>m2;
                        cout<<"\nEnter the mark 3:";
                        cin>>m3;
                        cout<<"\nEnter the mark 4:";
                        cin>>m4;
                        cout<<"\nEnter the mark 5:";
                        cin>>m5;
                }
};


class result: public student, public marks   //Deriving new class from more two base class
{
        private:
                int total;                    //Data Member Declaration
                float average;
                char result[6];

         public:

                voidcalculation()             //Function to calculate result of thestudent
                {
                        total=m1+m2+m3+m4+m5;
                        average=float(total)/5;
                        if(m1>49 && m2>49 && m3>49 && m4>49 && m5>49)
                        {
                                strcpy(result,"Pass");
                        }
                        else
                        {
                                strcpy(result,"Fail");
                        }
                }
```

```
		void display()		//Function to display student details
		{
			cout<<"\n\t\t\tSTUDENT DETAILS";
			cout<<"\nRegister Number:"<<regno;
			cout<<"\nName:"<<name;
			cout<<"\nDepartment:"<<dept;
			cout<<"\nMark 1:"<<m1;
			cout<<"\nMark 2:"<<m2;
			cout<<"\nMark 3:"<<m3;
			cout<<"\nMark 4:"<<m4;
			cout<<"\nMark 5:"<<m5;
			cout<<"\nTotal:"<<total;
			cout<<"\nAverage:"<<average;
			cout<<"\nResult:"<<result;
		}
};

int main()
{
	clrscr();
	result r; //Object declaration for derived class

	cout<<"\n\t\t\t STUDENT INFORMATION";
	r.getdata(); //Calling Base class function using derived object
	r.getmarks(); //Calling Base class function using derived object
	r.calculation();
	r.display();

	getch();
	return 0;
}
```

INPUT & OUTPUT:



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
Enter the department:cse

Enter the mark 1:90

Enter the mark 2:80

Enter the mark 3:70

Enter the mark 4:90

Enter the mark 5:80
                        STUDENT DETAILS
Register Number:101
Name:Mani
Department:cse
Mark 1:90
Mark 2:80
Mark 3:70
Mark 4:90
Mark 5:80
Total:410
Average:82
Result:Pass_
```

RESULT:

Thus the student information system using multiple inheritance executed and verified successfully

Ex. No: 4(B)	**MULTILEVEL INHERITANCE**	**Date:**

AIM:

To write a C++ program to implement Multilevel inheritance using Student information system

ALGORITHM:

1. Start the program.
2. Declare the base class student.
3. Declare and define the function get data() to get the student details.
4. Declare the other class marks.
5. Declare and define the function get marks() to get the student marks.
6. Create the another class result from the derived new class marks.
7. Declare and define the function calculation() to find out the total and average of the student.
8. Declare the function display() to display the student details.
9. Declare the derived class object and call the functions get data(), get marks(), get calculation() and display().
10. Stop the program

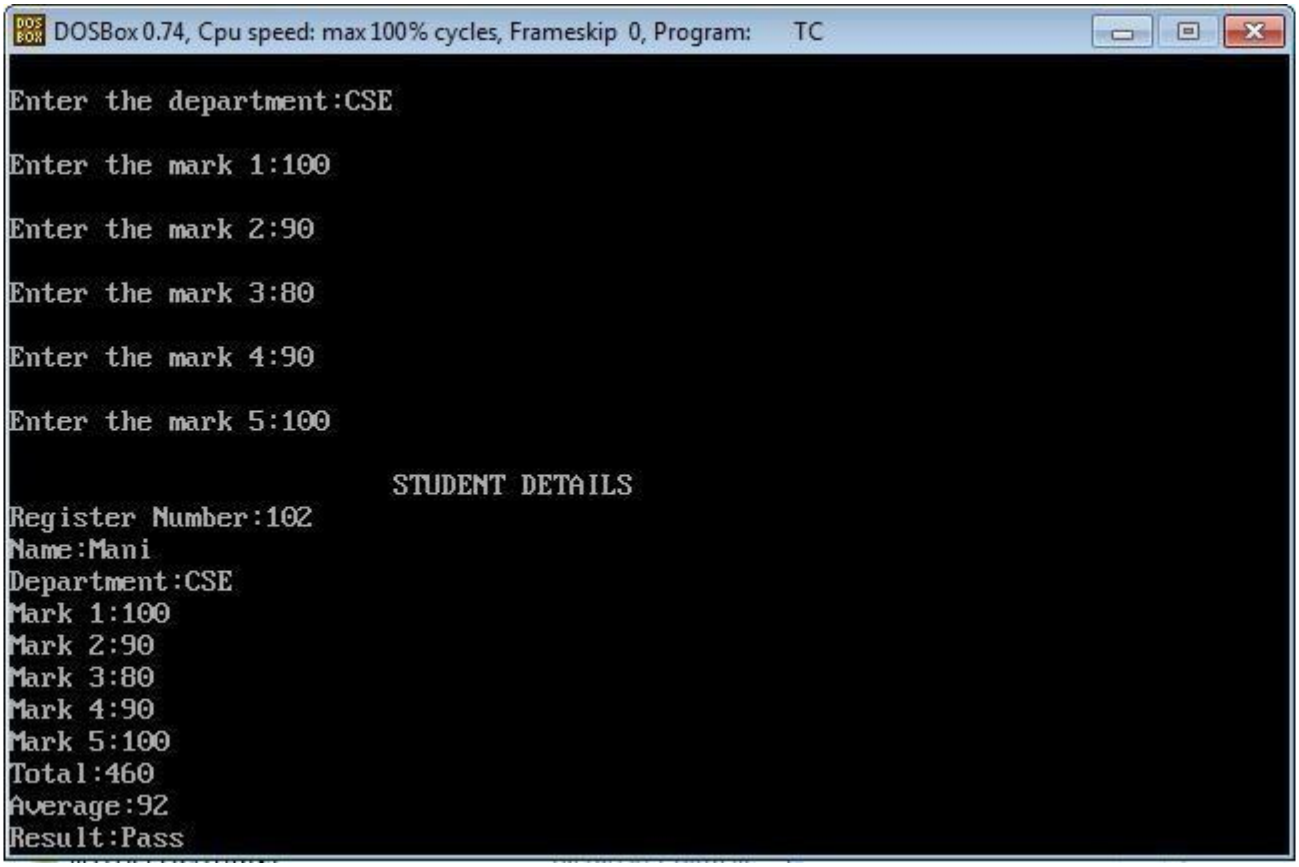
CODING:

```
#include<iostream.h>
#include<conio.h>
#include<string.h>


classstudent                                        //ClassDeclaration
{
  protected:
              intregno;                             //Data MemberDeclaration
              char name[20],dept[5];

  public:
              voidgetdata()                         //Function to get studentdetails
              {
                     cout<<"\nEnter the registernumber:";
                     cin>>regno;
                     cout<<"\nEnter the name:";
                     cin>>name;
                     cout<<"\nEnter the department:";
                     cin>>dept;
              }
};
```

```
class marks:public student              //Deriving new class from base class
{
  protected:
                int m1,m2,m3,m4,m5; //Data Member Declaration

  public:
                void get marks()        //Function to get student marks
                {
                        cout<<"\nEnter the mark 1:";
                        cin>>m1;
                        cout<<"\nEnter the mark 2:";
                        cin>>m2;
                        cout<<"\nEnter the mark 3:";
                        cin>>m3;
                        cout<<"\nEnter the mark 4:";
                        cin>>m4;
                        cout<<"\nEnter the mark 5:";
                        cin>>m5;
                }
};

class result:public marks      //Deriving new class from another derived class
{
  private:
                int total;          //Data Member Declaration
                float average;
                char result[6];

  public:

                void calculation()      //Function to calculate result of the student
                {
                        total=m1+m2+m3+m4+m5;
                        average=float(total)/5;
                        if(m1>49 && m2>49 && m3>49 && m4>49 && m5>49)
                        {
                                strcpy(result,"Pass");
                        }
                        else
                        {
                                strcpy(result,"Fail");
                        }
                }

                void display()                                  //Function to display student details
                {
                        cout<<"\n\t\t\tSTUDENT DETAILS";
```

```
            cout<<"\nRegister Number:"<<regno;
            cout<<"\nName:"<<name;
            cout<<"\nDepartment:"<<dept;
            cout<<"\nMark 1:"<<m1;
            cout<<"\nMark 2:"<<m2;
            cout<<"\nMark 3:"<<m3;
            cout<<"\nMark 4:"<<m4;
            cout<<"\nMark 5:"<<m5;
            cout<<"\nTotal:"<<total;
            cout<<"\nAverage:"<<average;
            cout<<"\nResult:"<<result;
        }
};

int main()
{
    clrscr();
    result r; //Object declaration for derived class

    cout<<"\n\t\t\t STUDENT INFORMATION";
    r.getdata();     //Calling Base class function using derived object
    r.getmarks();    //Calling Base class function using derived object
    r.calculation();
    r.display();

    getch();
    return 0;
}
```

INPUT & OUTPUT:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
Enter the department:CSE

Enter the mark 1:100

Enter the mark 2:90

Enter the mark 3:80

Enter the mark 4:90

Enter the mark 5:100

                        STUDENT DETAILS
Register Number:102
Name:Mani
Department:CSE
Mark 1:100
Mark 2:90
Mark 3:80
Mark 4:90
Mark 5:100
Total:460
Average:92
Result:Pass
```

RESULT:

Thus the student information system using multilevel inheritance executed and verified successfully

Ex. No: 4(C)	**HYBRID INHERITANCE**	**Date:**

AIM:

To write a C++ program to implement Hybrid inheritance using Student information system

ALGORITHM:

1. Start the program.
2. Declare the base class student.
3. Declare and define the function get data() to get the student details.
4. Declare and define the function show data() to display the student details.
5. Declare the other class marks as virtual.
6. Declare and define the function get marks() to get the student marks.
7. Declare and define the function show mark() to display mark
8. Create the another class attendance and make it virtual.
9. Declare and define the function get total hour() to find out the total hour.
10. Declare and define the function get absent() to find out the total absent hour.
11. Create the another class result which publically inherited classes marks and attendance
12. Declare the member function show calculation() which show the total marks and percentage
13. Declare the member function member function show display() to display the student details
14. Declare the function result().
15. Declare the derived class object, and call the functions get data(), get marks(), get attendance(), get calculation() and get display().
16. Stop the program

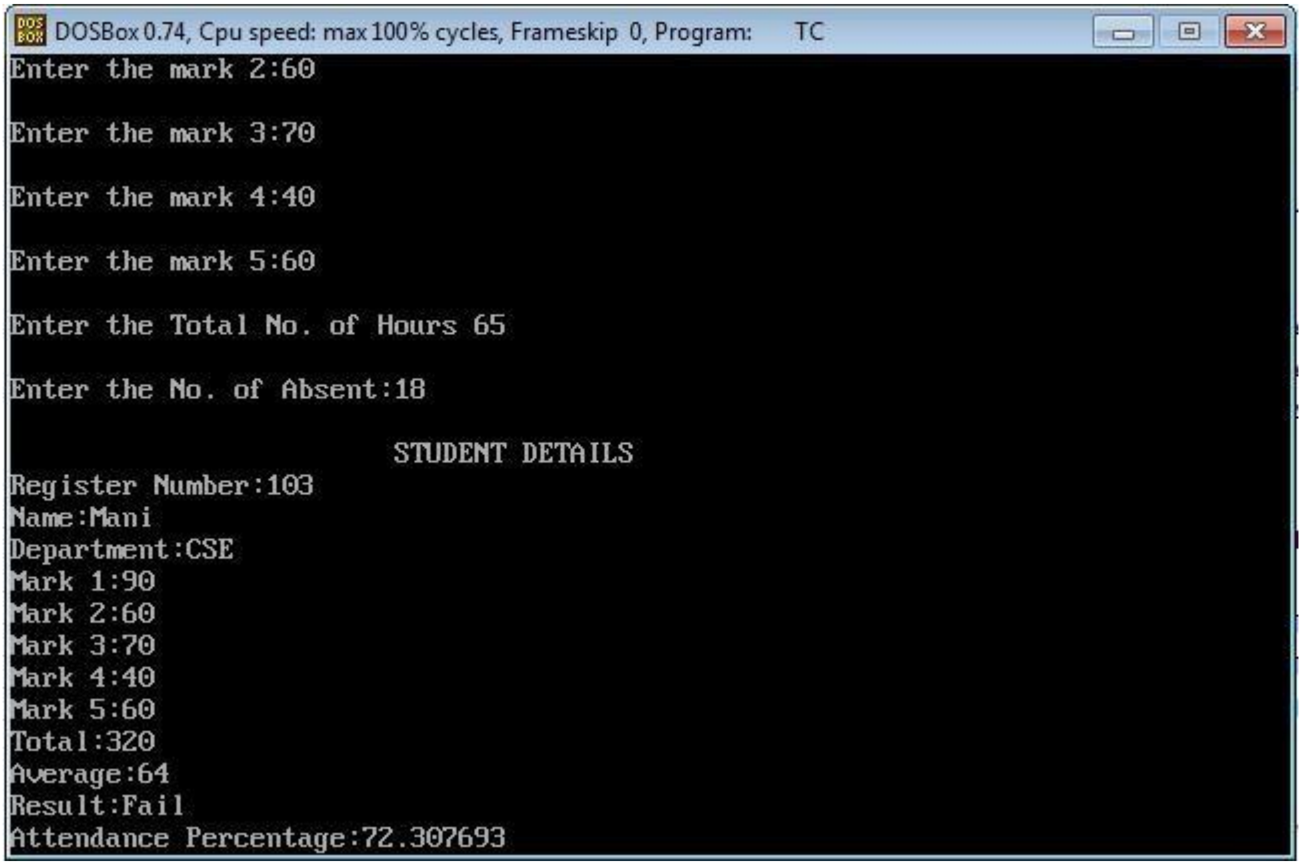
CODING:

```
#include<iostream.h>
#include<conio.h>
#include<string.h>

Class student                       //Class Declaration
{
  protected:
                int regno;          //Data Member Declaration
                char name[20],dept[5];

  public:
                voidgetdata()               //Function to get student details
                {
                        cout<<"\nEnter the register number:";
                        cin>>regno;
                        cout<<"\nEnter the name:";
                        cin>>name;
                        cout<<"\nEnter the department:";
                        cin>>dept;
                }
};

class marks: public student             //Deriving new class from another derived class
{
  protected:
                intm1,m2,m3,m4,m5;                      //Data Member Declaration

  public:
                void get marks()                //Function to get student marks
                {
                        cout<<"\nEnter the mark 1:";
                        cin>>m1;
                        cout<<"\nEnter the mark 2:";
                        cin>>m2;
                        cout<<"\nEnter the mark 3:";
                        cin>>m3;
                        cout<<"\nEnter the mark 4:";
                        cin>>m4;
                        cout<<"\nEnter the mark 5:";
                        cin>>m5;
                }
};

Class attendance                //Class Declaration
{
  protected:
                int total hours,absent,present;         //Data Member Declaration
                float percentage;
  public:
                void get attendance()           //Function to get student attendance details
                {
                        cout<<"\nEnter the Total No. of Hours ";
                        cin>>total hours;
                        cout<<"\nEnter the No. of Absent:";
                        cin>>absent;
                }
};


class result: public marks,public attendance
{
  private:
                int total; //Data Member Declaration
                float average;
```

```
        char result[6];

  public:

    void calculation()      //Function to calculate result of the student
        {
            total=m1+m2+m3+m4+m5;
            average=float(total)/5;
            if(m1>49 && m2>49 && m3>49 && m4>49 && m5>49)
            {
                strcpy(result,"Pass");
            }
            else
            {
                strcpy(result,"Fail");
            }
            present=total hours-absent;
            percentage=float(present)/total hours;
            percentage=percentage*100;

        }

        void display()//Function to display student details
        {
            cout<<"\n\t\t\tSTUDENT DETAILS";
            cout<<"\nRegister Number:"<<regno;
            cout<<"\nName:"<<name;
            cout<<"\nDepartment:"<<dept;
            cout<<"\nMark 1:"<<m1;
            cout<<"\nMark 2:"<<m2;
            cout<<"\nMark 3:"<<m3;
            cout<<"\nMark 4:"<<m4;
            cout<<"\nMark 5:"<<m5;
            cout<<"\nTotal:"<<total;
            cout<<"\nAverage:"<<average;
            cout<<"\nResult:"<<result;
            cout<<"\nAttendance Percentage:"<<percentage;
        }
};
int main()
{
    clrscr();
    result r;   //Object declaration

    cout<<"\n\t\t\t STUDENT INFORMATION";
    r.get data();
    r.get marks();
```

```
        r. getattendance();
        r.calculation();
        r.display();

        getch();
        return 0;
}
```

INPUT & OUTPUT:



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
Enter the mark 2:60

Enter the mark 3:70

Enter the mark 4:40

Enter the mark 5:60

Enter the Total No. of Hours 65

Enter the No. of Absent:18

                        STUDENT DETAILS
Register Number:103
Name:Mani
Department:CSE
Mark 1:90
Mark 2:60
Mark 3:70
Mark 4:40
Mark 5:60
Total:320
Average:64
Result:Fail
Attendance Percentage:72.307693
```

RESULT:

Thus the student information system using hybrid inheritance executed and verified successfully

Ex. No: 5	**ABSTRACT CLASS**	**Date:**

AIM:

To write a java program to demonstrate the use of abstract classes.

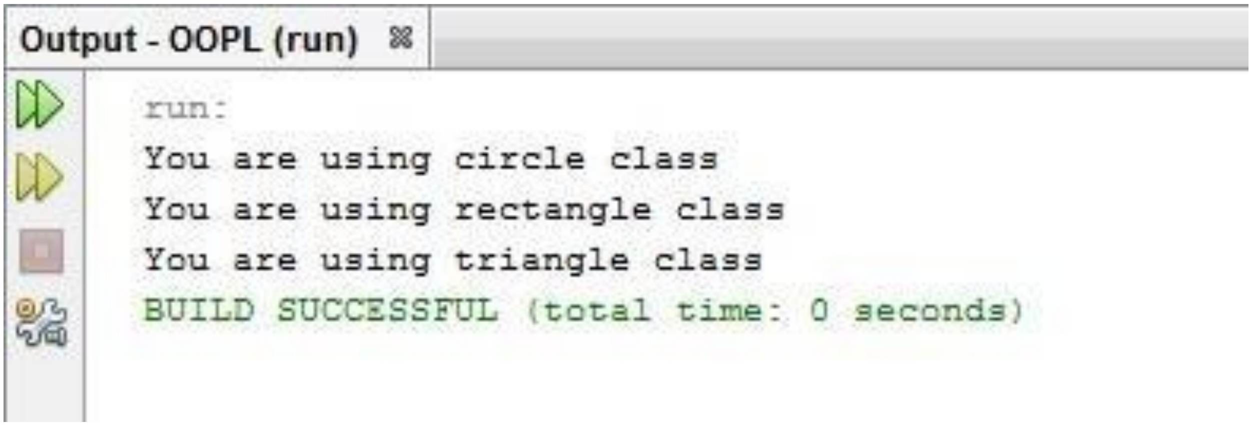
ALGORITHM:

1. Create an abstract class Shape
2. Declare display() as abstract method
3. Create circle class by extending shape class and override the display()method
4. Create rectangle class by extending shape class and override the display()method
5. Create triangle class by extending shape class and override the display()method
6. Create circle object and assign it to shape object reference and call display method
7. Createrectangleobjectandassignittoshapeobjectreferenceandcalldisplaymethod
8. Createtriangleobjectandassignittoshapeobjectreferenceandcalldisplaymethod
9. Stop the program.

CODING:

```
abstract class Shape
{
   abstract void display();

}

class Circle extends Shape
{
   void display()
   {
        System.out.println("You are using circle class");
   }
}

class Rectangle extends Shape
{
   void display()
   {
        System.out.println("You are using rectangle class");
   }
}

class Triangle extends Shape
{
   void display()
{

            System.out.println("You are using triangle class");
      }
    }

    class ShapeAbstract
    {
        public static void main(String args[])
        {
            Shape s = new Circle();
            s.display();

            s = newRectangle();
            s.display();

            s = new Triangle();
            s.display();
        }
    }
```

OUTPUT



```
run:
You are using circle class
You are using rectangle class
You are using triangle class
BUILD SUCCESSFUL (total time: 0 seconds)
```

:

RESULT:

Thus the use of abstract class program executed and verified successfully

Ex. No: 6(A)	**I/O STREAMS**	**Date:**

AIM:

To write a java program to demonstrate simple i/o streams and its methods

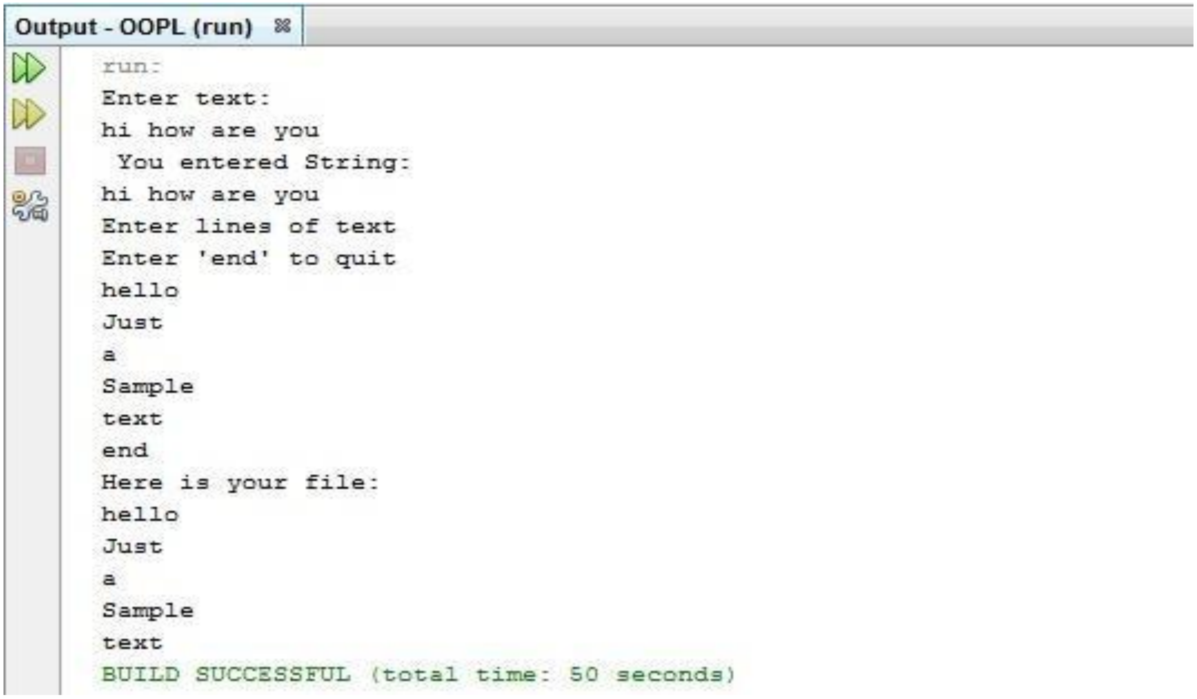
ALGORITHM:

1. Create class IODemo
2. Create objects for Input StreamReader and Buffered Reader
3. With read Line() method, get input from the console and store it in string str
4. Declare String array str1 with size of100
5. Using for loop, get multiple lines as input using read Line()method
6. The loop will terminate if the string matches with “end”
7. Display the str1array
8. Stop the program

CODING:

```
import java.io.*;

public class IODemo
{
   public static void main(String[] args)
   {
      try
      {
         InputStreamReader ir=new InputStreamReader(System.in);
         BufferedReader br=new BufferedReader(ir);

         //single line as input
         System.out.println("Enter text: ");
         String str= br.readLine();
         System.out.println(" You entered String: ");
         System.out.println(str);

         //multiple lines as input
         String str1[]=newString[100];
         System.out.println("Enter lines of text");
         System.out.println("Enter 'end' to quit");

         for(int i=0; i<100; i++)
         {
            str1[i]=br.readLine();
            if(str1[i].equals("end"))
               break;
         }
         System.out.println("Here is your file:");
         for(int i=0; i<100; i++)
         {
            if(str1[i].equals("end"))
               break;
            System.out.println(str1[i]);
         }
      }
      catch(IOException e)
      {
         System.out.println(e);
      }
   }
}
```

INPUT/OUTPUT:



```
run:
Enter text:
hi how are you
 You entered String:
hi how are you
Enter lines of text
Enter 'end' to quit
hello
Just
a
Sample
text
end
Here is your file:
hello
Just
a
Sample
text
BUILD SUCCESSFUL (total time: 50 seconds)
```

RESULT:

Thus the i/o stream demonstration program executed and verified successfully

Ex. No: 6(B)	**I/O STREAMS - FILE**	**Date:**

AIM:

To write a java program to demonstrate file i/o streams and its methods

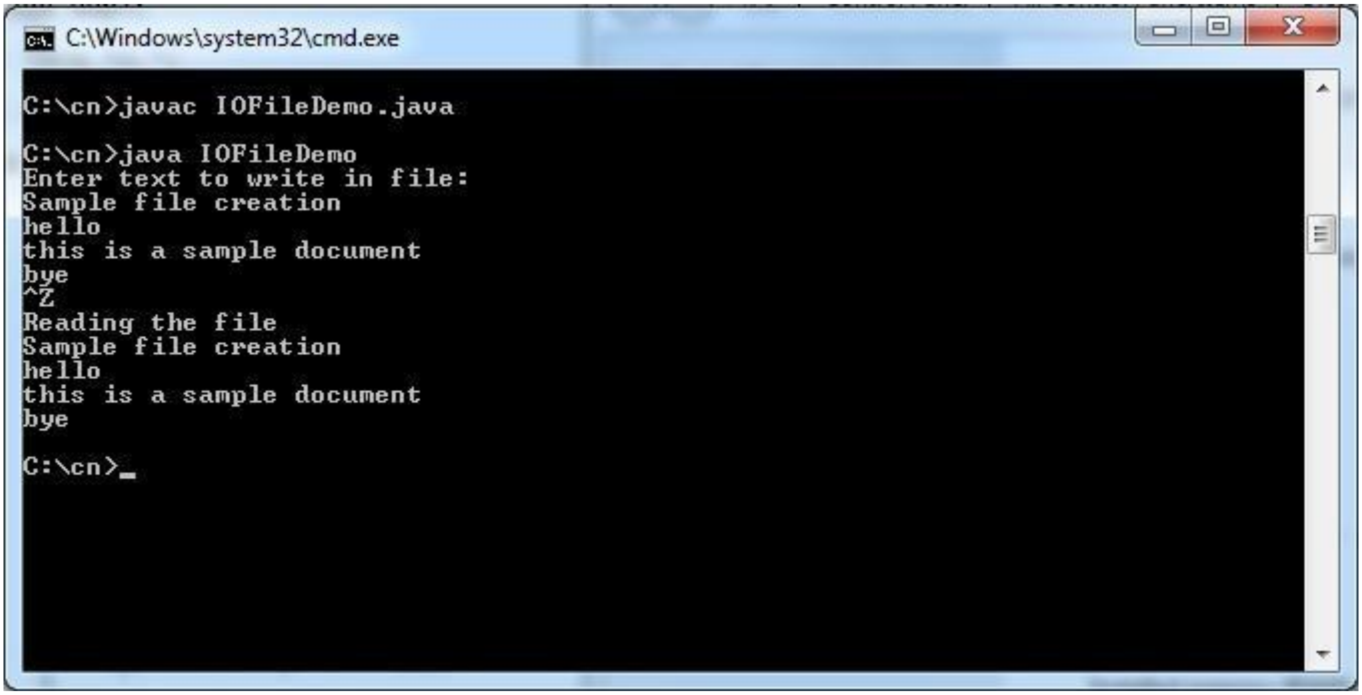
ALGORITHM:

1. Create class IO File Demo
2. Declare objects for Input Stream Reader and Buffered Reader
3. Declare an object for File Output Stream to perform write operation in a file
4. Using br. read() method, get input from the console
5. Then write the input to the file using f out object
6. While completes writing the file, close the f out stream
7. Declare an object for File Input Stream to perform read operation in a file
8. Declare objects for Input Stream Reader and Buffered Reader associated with the file object
9. Using read Line() method, read all the contents in the file and display it on the console
10. Close the f is stream
11. Stop the program

CODING:

```
import java.io.*;

public class IOFileDemo
{
   public static void main(String args[])
   {
      InputStreamReader ir=new InputStreamReader(System.in);
      BufferedReader br=new BufferedReader(ir);
      System.out.println("Enter text to write in file:");
      int i;

     //To perform write operation in a file
      try
      {
         FileOutputStream fout=new FileOutputStream("sample.txt");
         do
         {
            i=br.read();
            if(i!=-1)
               fout.write(i);
         }while(i!=-1);
      fout.close();
    }

    catch(IOException e)
    {
      System.out.println(e);
    }


    //to perform read operation in a file
    try
    {
      File Input Stream fis=new File Input Stream("sample.txt");
      Data Input Stream dis=new Data Input Stream(fis);
      Buffered Reader br1=newBufferedReader(new InputStreamReader(dis));

      String str;

      System.out.println("Reading the file");

     while((str=br1.readLine())!=null)
      {
        System.out.println(str);
      }

       dis.close();
    }

    catch(IOException e)
    {
      System.out.println(e);
    }

  }
}
```

INPUT/OUTPUT:



```
C:\cn>javac IOFileDemo.java

C:\cn>java IOFileDemo
Enter text to write in file:
Sample file creation
hello
this is a sample document
bye
^Z
Reading the file
Sample file creation
hello
this is a sample document
bye

C:\cn>_
```

RESULT:

Thus the demonstration of file i/o stream program executed and verified successfully

Ex. No: 7	**TYPE CONVERSION**	**Date:**

AIM:

To write a java program to perform all possible type conversions.

ALGORITHM:

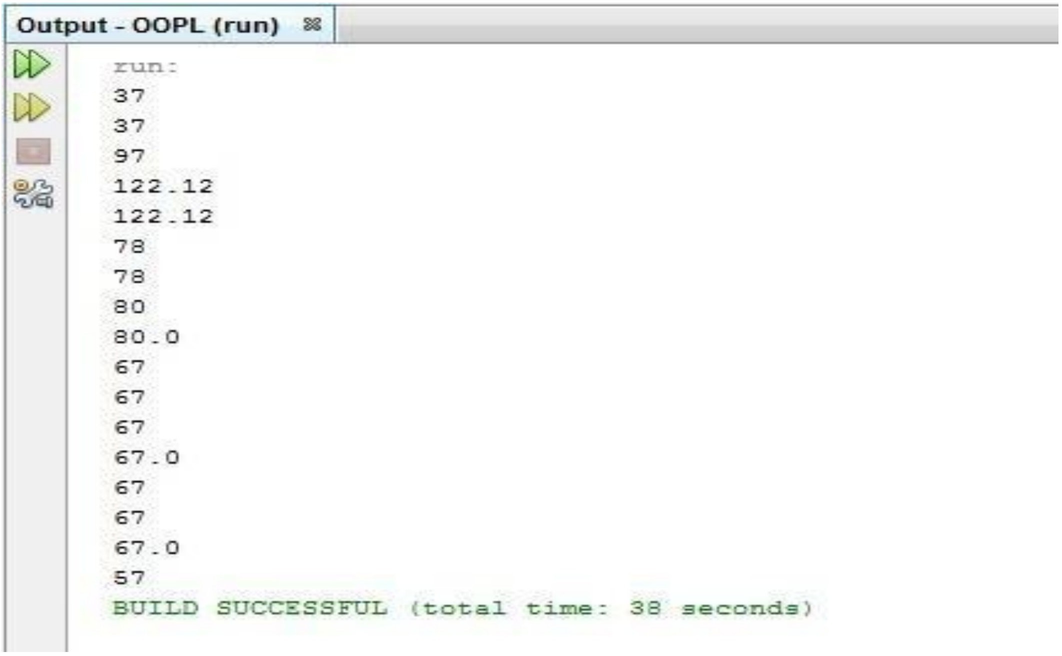
1. Create a class Type Conversion
2. Import java.io package to use io streams and methods
3. Created object for scanner class to get input from console
4. Convert string to integer using Integer .parse Int()method
5. Convert integer to string using Integer.to String()method
6. Convert xxx data type to string using xxx.to String()method
7. Convert string to xxx data type using xxx .parse xxx()method
8. Convert string to other types using value Of()method
9. Convert char to int using external typecasting
10. Stop the program

CODING:

```
import java.io.*;
import java.lang.*;
import java.util.*;

public class TypeConversion
{
   public static void main(String arg[])
   {
      Scanner in=new Scanner(System.in);

      //String to Integer
      String s=in.next();
      int i=Integer.parseInt(s);
      System.out.println(i);

      //Integer to String
      int a=97;
      String s3=Integer.toString(a);
      System.out.println(s3);

      //Double to String
      double d=in.nextDouble();
      s=Double.toString(d);
      System.out.println(s);
```

```
        //Long to String
        long l=in.nextLong();
        s=Long.toString(l);
        System.out.println(s);

        //Float to String
        float f=in.nextFloat();
        s=Float.toString(f);
        System.out.println(s);

        //String to Integer
        s=in.next();
        i=Integer.valueOf(s).intValue();
        System.out.println(i);
        i =Integer.parseInt(s);
        System.out.println(i);

        //String to Double
        d=Double.valueOf(s).doubleValue();
        System.out.println(d);

        //String to Long
        long lng=Long.valueOf(s).longValue();
        System.out.println(lng);
        lng=Long.parseLong(s);
        System.out.println(lng);

        //String to Float
        f=Float.valueOf(s).floatValue();
        System.out.println(f);

        //Character to Integer
        char c='9';
        i=(char)c;
        System.out.println(i);


    }
}
```

INPUT & OUTPUT:



```
run:
37
37
97
122.12
122.12
78
78
80
80.0
67
67
67
67.0
67
67
67.0
57
BUILD SUCCESSFUL (total time: 38 seconds)
```

RESULT:

Thus the type conversion program executed and verified successfully

Ex. No: 8(a)	**EXCEPTION HANDLING – DIVIDE BY ZERO**	**Date:**

AIM:

To write a java program to demonstrate exception handling technique.

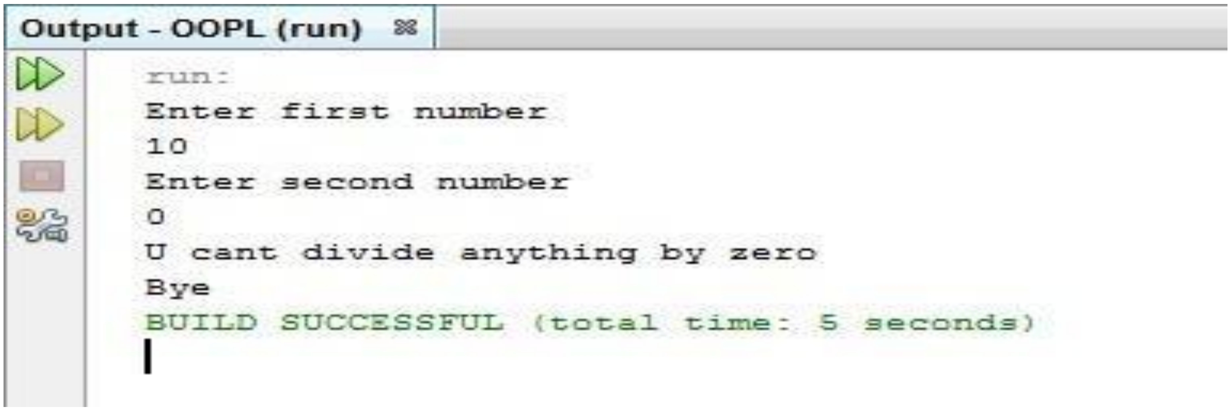
ALGORITHM:

1. Create a classException.
2. Try dividing a numeric value by 0 in the tryblock
3. Catch the exception as Arithmetic exception and print the result as "Divide by Zero error".
4. Finally compile and run the program to verify theresult.

CODING:

```
//Program to demonstrate Exception handling in java
class exception
{
        public static void main(String args[ ])
        {
                int a=10,b=5,c=5;
                int x,y;
                try
                {
                        x = a/(b-c);
                }
                catch(ArithmeticException e)
                {
                        System.out.println("Division by zero");
                }
                System.out.println("Program End");
        }
}
```

INPUT & OUTPUT:



```
run:
Enter first number
10
Enter second number
0
U cant divide anything by zero
Bye
BUILD SUCCESSFUL (total time: 5 seconds)
```

RESULT:

Thus the exception handling program executed and verified successfully

Ex. No: 8(B)	**EXCEPTION HANDLING – MULTIPLE CATCH BLOCK**	**Date:**

AIM:

To write a java program to demonstrate multiple catch block in exception handling technique.

ALGORITHM:

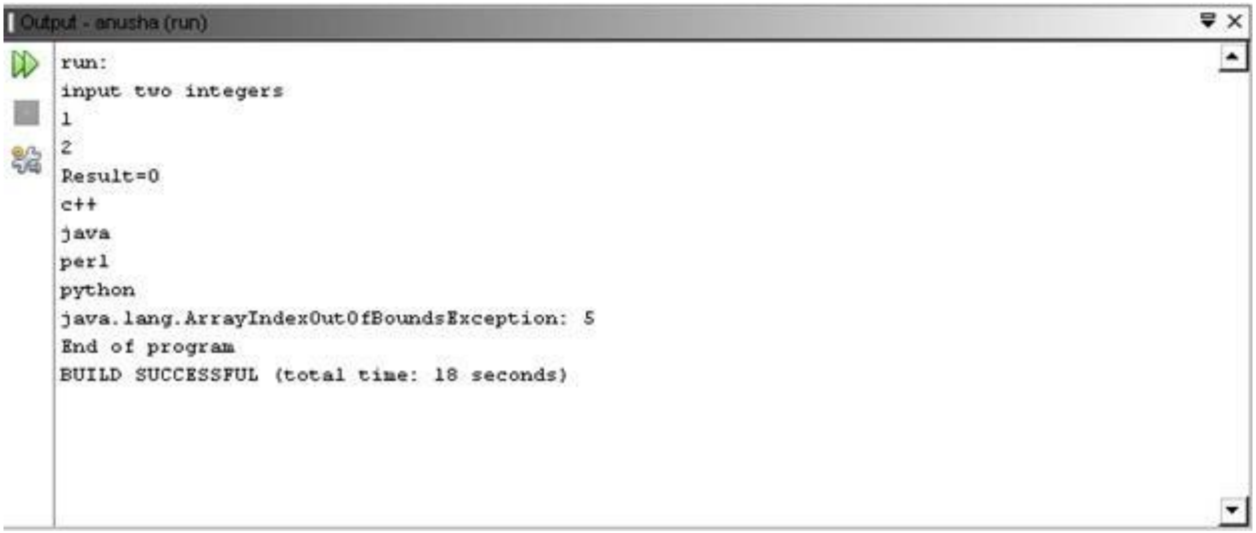
1. Create a class Multiple Catch Block Demo with the method check bound()
2. Create a string array 'Language' and initialize it with the values.
3. Try printing the values exceeding the array bounds, the compiler displays the array index out of bounds exception if the exception is caught.
4. Similarly try dividing a value by0.
5. Multiple catch statements can be used to catch the exceptions such as Arithmetic exception, Divide by zero exception ,etc.
6. Compile and run the program to verify the exception concept

CODING:

```
import java. util. Scanner;
class Multiple Catch BlockDemo
{
        static void check bound() throws Array IndexOutOfBoundsException
        {
            String languages[] = { "C", "C++", "Java", "Perl", "Python" };
            for (int c = 1; c <= 5; c++)
            {
                System.out.println(languages[c]);
            }
        }

        public static void main(String[] args)
        {
            int a, b, result;

            Scanner input = new Scanner(System.in);
            System.out.println("Input two integers");

            a = input.nextInt();
            b = input.nextInt();

            // try block
            try
            {
```

```
                        result = a / b;
                        System.out.println("Result = " + result);
                        checkbound();
                }

                //Multiple catch blocks
                catch (ArithmeticException e)
                {
                        System.out.println("Exception caught: Division by zero.");
                }
                catch (ArrayIndexOutOfBoundsException e)
                {
                        System.out.println(e);
                }

                catch (Exception e)
                {
                        System.out.println("Exception caught:" + e.getmessage());
                }

                System.out.println("End of the Program");

        }
}
```

INPUT & OUTPUT:



```
run:
input two integers
1
2
Result=0
c++
java
perl
python
java.lang.ArrayIndexOutOfBoundsException: 5
End of program
BUILD SUCCESSFUL (total time: 18 seconds)
```

RESULT:

Thus the exception handling with multiple catch blocks program executed and verified successfully

Ex. No: 9	NETWORKING – TCP ONE WAY COMMUNICATION	**Date:**

AIM:

To write a java program for one way communication using TCP

ALGORITHM:

Server Program:

1. Start.
2. Import the java.net and java.io packages.
3. Declare a new class called "Tcp One Way Chat Server".
4. Within class "Tcp One Way Chat Server", create a Server Socket object called "ss" with the port number 8000.
5. Create a Socket object called "s" by using the accept()method ,which listens for a connection to be made to this server socket and accepts it.
6. Create a new Buffered Reader object, which acts as an input stream to the server from the client.
7. Create a new Print Stream object, which acts as an output stream to the client from the server.
8. Display the message "Server ready..." on the server window.
9. Repeat the following steps:
   i. Prompt for the message to be sent from server to client.
   ii. Read in the message to be sent from the user.
   iii. Send the message to the client using the Print Stream object.
   iv. If the message equals the string "end", then close the input and output streams andexit the loop.
10. Stop

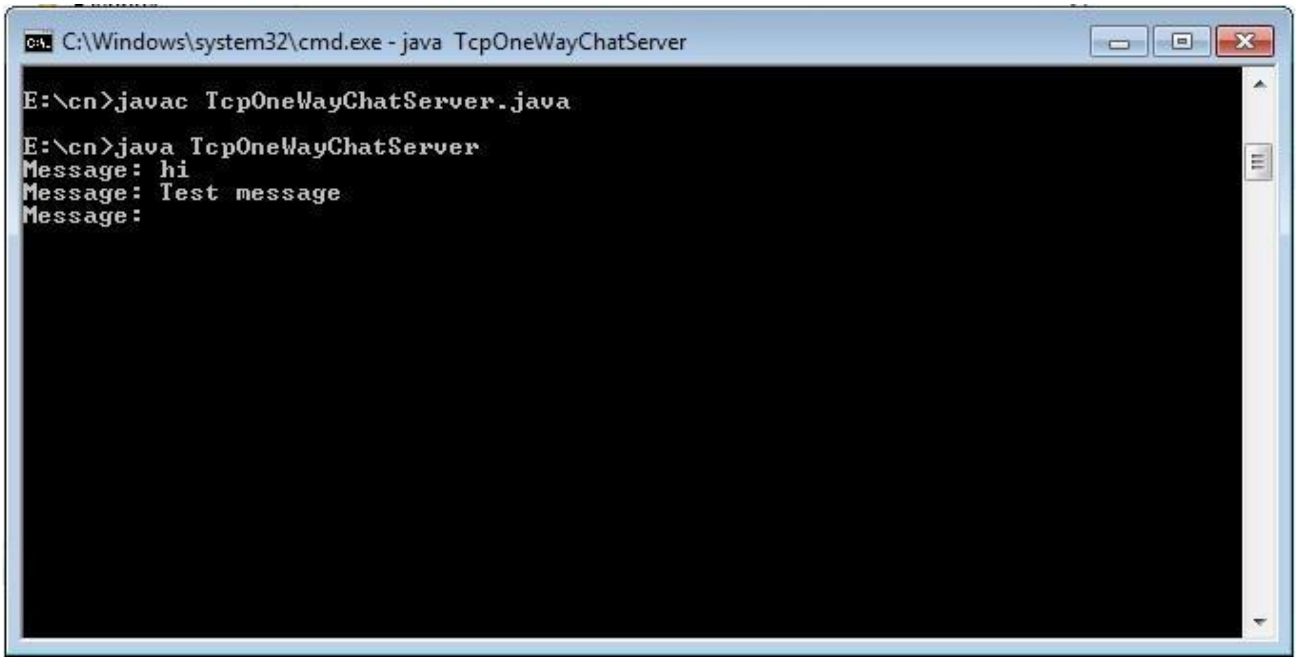
Client Program:

1. Start.
2. Import the java.net, java.io, and java.utilpackages.
3. Create a new class called"Tcp One WayClient".
4. Inside "Tcp One Way Client", create a new Socket object called "s" with port number8000.
5. Create a new Buffered Reader object which acts as the input stream to the client from the server.
6. Repeat the following steps:
   i. Read in the input from the server to the client using the BufferedReaderobject.
   ii. Display the message received.
   iii. If the string received equals" end", then close the input and output streams and exit the loop.
7. Stop.

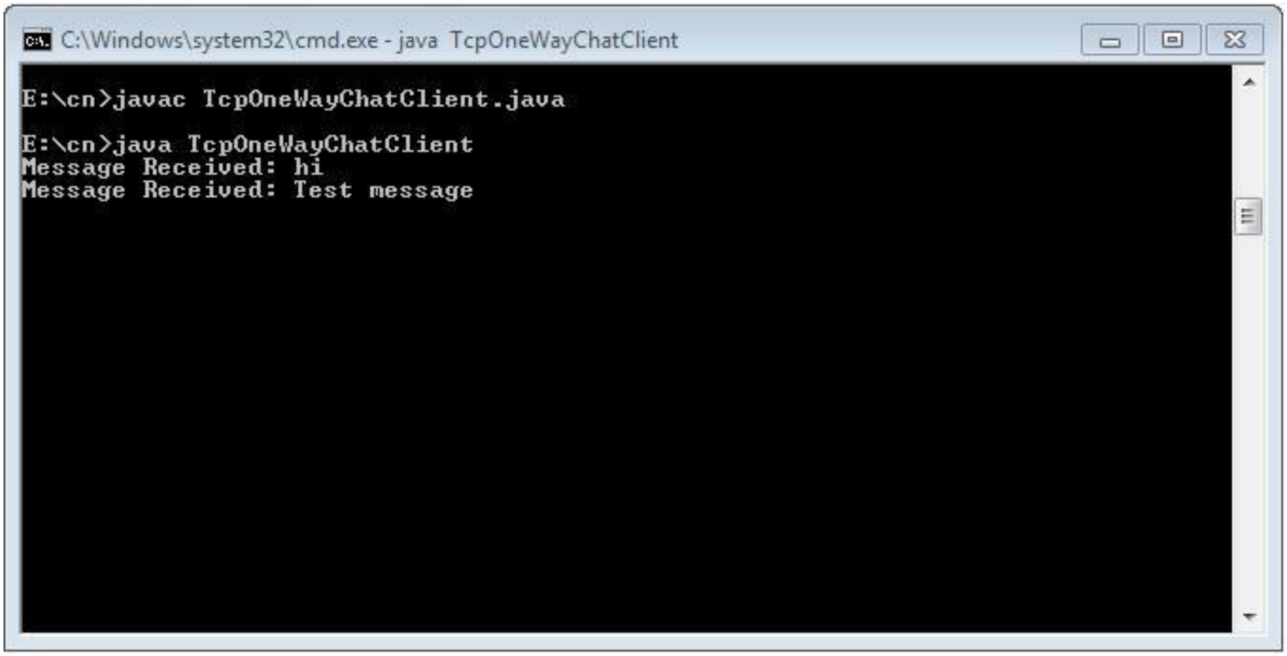
CODING

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

class TcpOneWayChatServer
{
    public static void main(String a[])throws IOException
    {
        ServerSocket ss = new ServerSocket(8000);
        Socket s=ss.accept();

        BufferedReader keyIn = new BufferedReader(new InputStreamReader(System.in));
        PrintStream socOut = new PrintStream(s.getOutputStream());

        while(true)
        {
            System.out.print("Message: ");
            String str = in.readLine();
            if(str.equals("bye"))
            {
                break;
            }
            socOut.println(str);
        }

    }
}

//TcpOneWayChatClient
import java.io.*;
import java.net.*;

class TcpOneWayChatClient
{
    public static void main(String args[])throws IOException
    {
        Socket c = new Socket("localhost", 8000);
        BufferedReader socIn=new BufferedReader(new
                     InputStreamReader(c.getInputStream()));
        String str;
        while(true)
        {
            str = socIn.readLine();
            System.out.println("Message Received: " + str);
        }
    } }
```

OUTPUT



```
E:\cn>javac TcpOneWayChatServer.java

E:\cn>java TcpOneWayChatServer
Message: hi
Message: Test message
Message:
```



```
E:\cn>javac TcpOneWayChatClient.java

E:\cn>java TcpOneWayChatClient
Message Received: hi
Message Received: Test message
```

RESULT:

Thus the one way communication using TCP program executed and verified successfully

Ex. No: 10	REMOTE METHOD INVOCATION	**Date:**

AIM:

To write a java program to design and implement RMI

ALGORITHM:

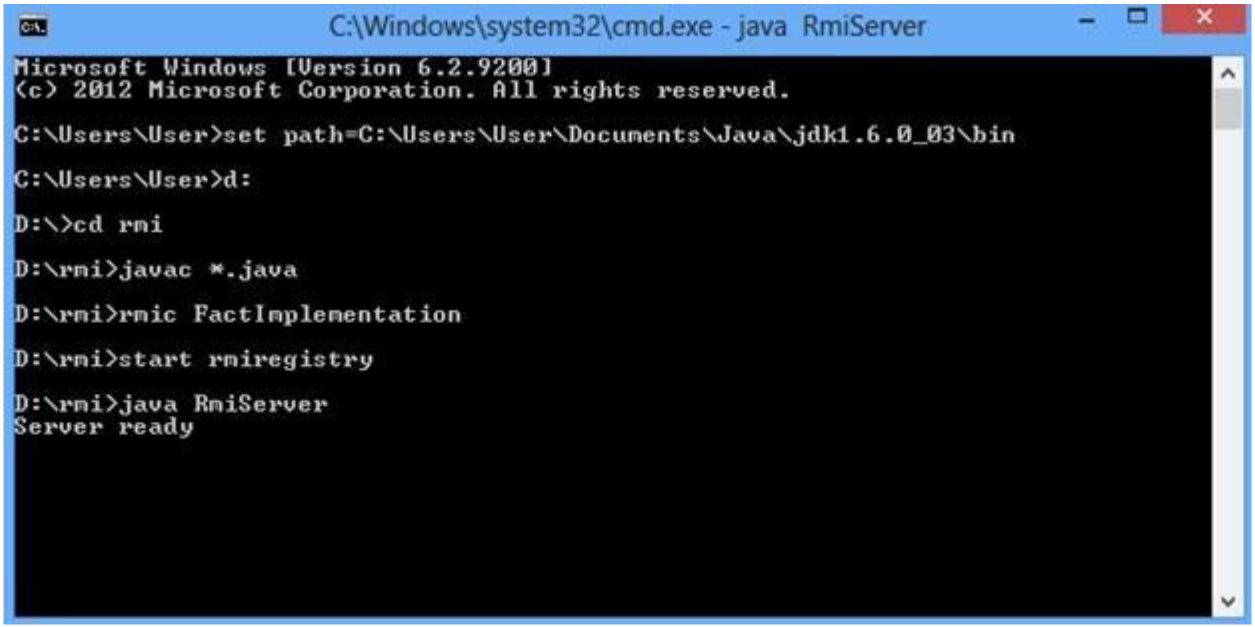
1. Create an interface named Fact Interface
2. Declare fact method in the Fact Interface
3. Create Fact Implementation class by implementing Fact Interface interface and define fact method
4. Create Rmi Server class and create an object for Fact Implementation class
5. Bind that object in the rmi registry using Naming. rebind() method and display server ready
6. Create Rmi Client class and lookup into the server's registry for the object using Naming. Look Up()method
7. Get the object from server and typecasting it as reference
8. Using that object, invoke fact method and display the output
9. Stop the program

CODING:

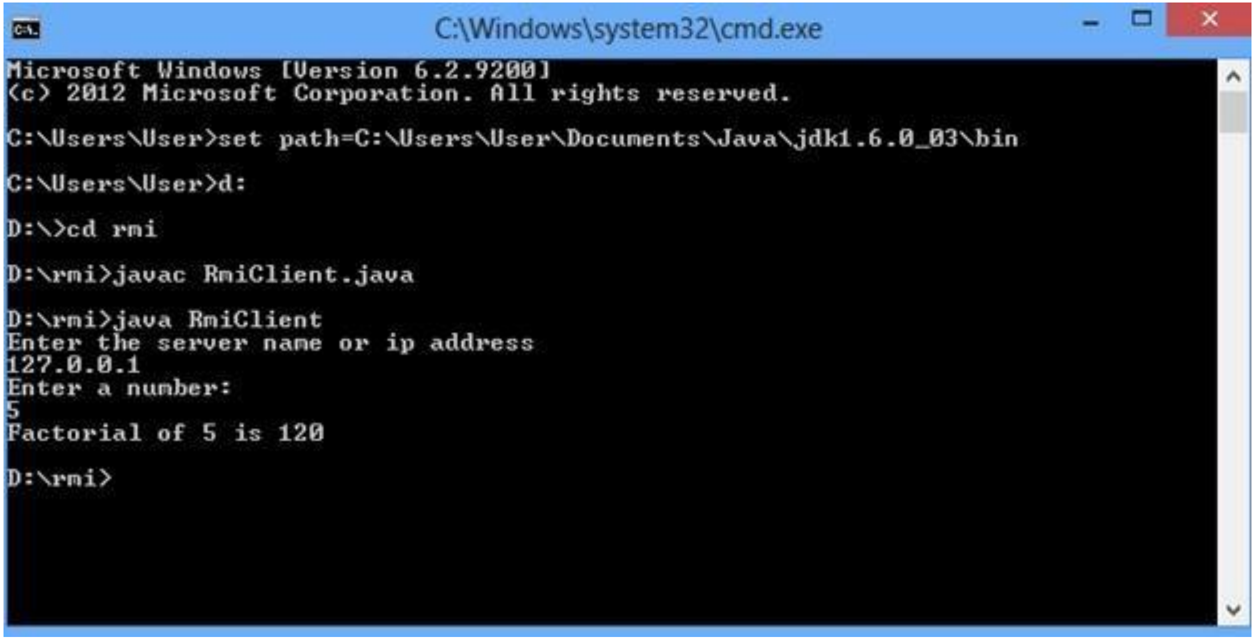
```
    //Fact Interface
    import java.rmi.*;
    import java.rmi.server.*;
    public interface FactInterface extends Remote
    {
       public int fact(int n) throws RemoteException;
    }

    //Fact Implementation
    import java.rmi.*;
    import java.rmi.server.*;
    public class FactImplementation extends UnicastRemoteObject implements FactInterface
    {
       public FactImplementation() throws RemoteException
       { }
       public int fact(int n) throws RemoteException
       {
          int f = 1;
          for(int i = 1; i <= n; i++)
          {
             f = f * i;
          }
          return f;
       }
}
```

```
//RMI Server
import java.rmi.*;
import java.net.*;
public class RmiServer
{
   public static void main(String args[]) throws RemoteException
   {
      try
      {
         FactImplementation fi = new FactImplementation();
         Naming.rebind("Server", fi);
         System.out.println("Server ready");
      }
      catch(Exception e)
      {
         System.out.println("Exception:" + e);
      }
   }
}

//RMI client
import java.rmi.*;
import java.io.*;
public class RmiClient
   {
   public static void main(String args[]) throws RemoteException
   {
      try
      {
         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
         System.out.println("Enter the server name or ip address");
         String url = br.readLine();
         String host="rmi://" + url + "/server";
         FactInterface serv = (FactInterface) Naming.lookup(host);
         System.out.println("Enter a number:");
         int n = Integer.parseInt(br.readLine());
         int fact = serv.fact(n);
         System.out.println("Factorial of " + n + " is " + fact);
      }
      catch(Exception e)
      { System.out.println("Error " + e); } } }
```

OUTPUT



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\User>set path=C:\Users\User\Documents\Java\jdk1.6.0_03\bin

C:\Users\User>d:

D:\>cd rmi

D:\rmi>javac *.java

D:\rmi>rmic FactImplementation

D:\rmi>start rmiregistry

D:\rmi>java RmiServer
Server ready
```



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\User>set path=C:\Users\User\Documents\Java\jdk1.6.0_03\bin

C:\Users\User>d:

D:\>cd rmi

D:\rmi>javac RmiClient.java

D:\rmi>java RmiClient
Enter the server name or ip address
127.0.0.1
Enter a number:
5
Factorial of 5 is 120

D:\rmi>
```

RESULT:

Thus the RMI using factorial method program executed and verified successfully

Ex. No: 11	**AWT CONCEPTS**	**Date:**

AIM:

To write a java program to implement AWT concepts.

ALGORITHM:

1. Create a Awt Control class which extends the Frame class
2. Set the grid layout with labels ,text boxes, dropdown boxes and checkboxes
3. Add two buttons, Show and Exit with the add Action Listener()method.
4. Then add the components such as labels, textboxes, dropdown boxes and checkboxes.
5. Perform the actions in the action Performed()method.
6. We can set the size of the layout using set Size()method.
7. Create the main() function and the object for the Awt Control class and call the methods using the objects.
8. Finally compile and run the program

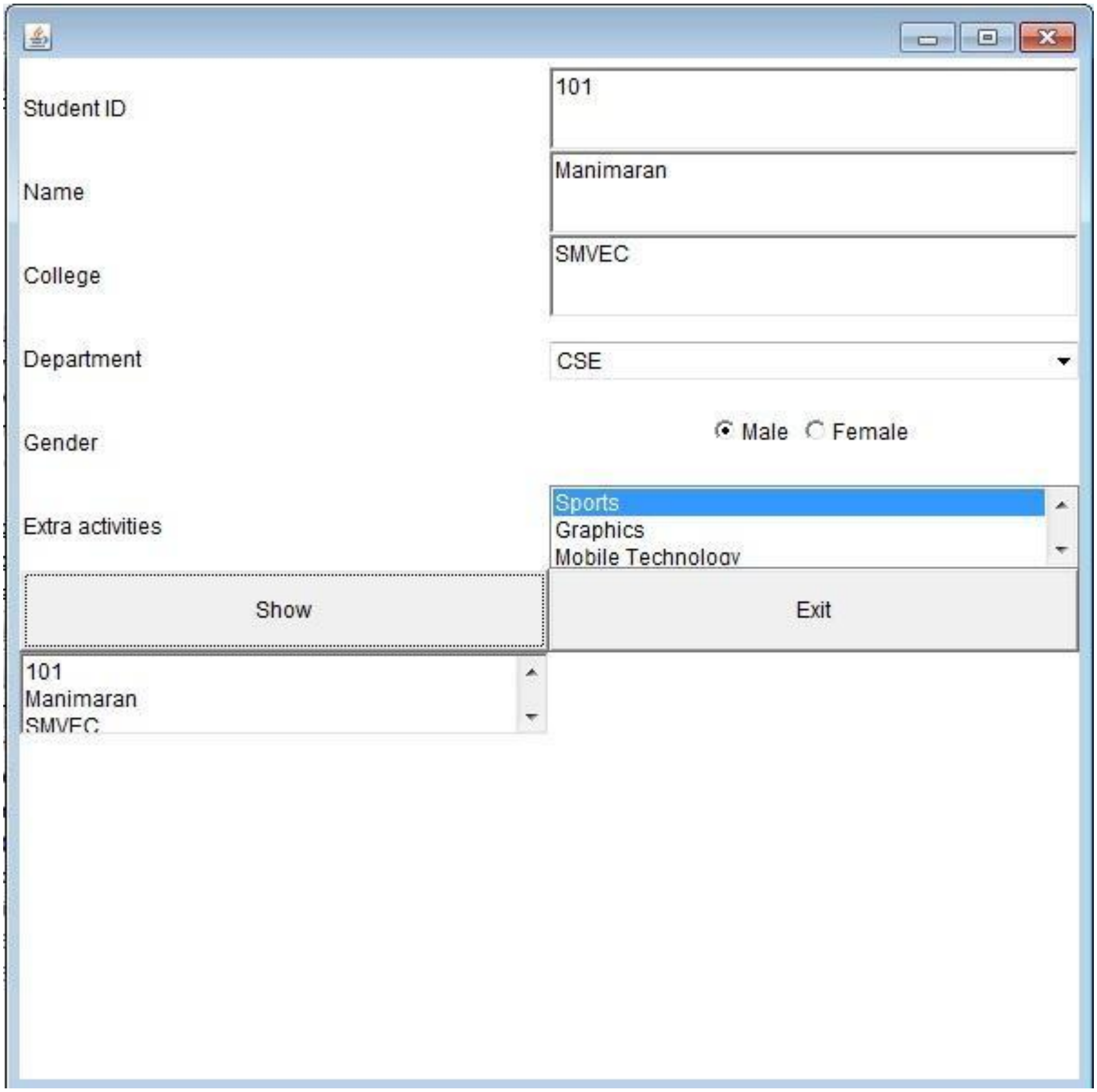
CODING:

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

public class AwtControl extends Frame implements Action Listener
{
    Label l1, l2, l3, l4, l5, l6, l7;
    TextField t1, t2, t3, t4;
    Choice ch1;
    Checkbox cb1, cb2;
    CheckboxGroup cbg;
    List lt;
    Button b1, b2;
    Panel p1;
    TextArea ta;

    public AwtControl()
    {
       setLayout(new GridLayout(12, 2));

       ta = new TextArea( 10, 20);

       l1 = new Label("Student ID ");
       l2 = new Label("Name ");
```

```
l3 = new Label("College");
l4 = new Label("Department");
l5 = new Label("Gender ");
l6 = new Label("Extra activities");

t1 = new TextField(20);
t2 = new TextField(20);
t3 = new TextField(20);

ch1 = new Choice();
ch1.addItem("CSE");
ch1.addItem("IT");
ch1.addItem("MCA");
ch1.addItem("ECE");

cbg = new CheckboxGroup();
cb1 = new Checkbox("Male", cbg, true);
cb2 = new Checkbox("Female", cbg, true);

p1 = new Panel();
p1.add(cb1);
p1.add(cb2);
It = newList();
It.addItem("Sports");
It.addItem("Graphics");
It.addItem("Mobile Technology");

b1 = new Button("Show");
b1.addActionListener(this);
b2 = new Button("Exit");
b2.addActionListener(this);

add(l1);
add(t1);
add(l2);
add(t2);
add(l3);
add(t3);
add(l4);
add(ch1);
add(l5);
```

```
        add(p1);
        add(l6);
        add(It);
        add(b1);
        add(b2);
        add(ta);

    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==b2)
        {
                System.exit(0);
        }
        ta.setText("");           //Clear
        ta.append(t1.getText() +"\n");
        ta.append(t2.getText()+ "\n");
        ta.append(t3.getText()+ "\n");
        ta.append(ch1.getSelectedItem()+ "\n");
        ta.append(cbg.getSelectedCheckbox().getLabel()+ "\n");
        ta.append(It.getSelectedItem()+ "\n");
    }

    public void windowClosing(WindowEvent e)
    {
          dispose();
          System.exit(0);
    }

    public static void main(String args[])
    {
        AwtControl control = new AwtControl();
        control.setSize(600,600);
        control.setVisible(true);
    }
}
```

INPUT & OUTPUT:



Student ID 101

Name Manimaran

College SMVEC

Department CSE

Gender ● Male ○ Female

Extra activities Sports
Graphics
Mobile Technology

```
101
Manimaran
SMVEC
```

RESULT:

Thus the AWT program executed and verified successfully

Ex. No: 12	**SWING CONCEPTS**	**Date:**

AIM:

To write a java program to implement swing concepts.

ALGORITHM:

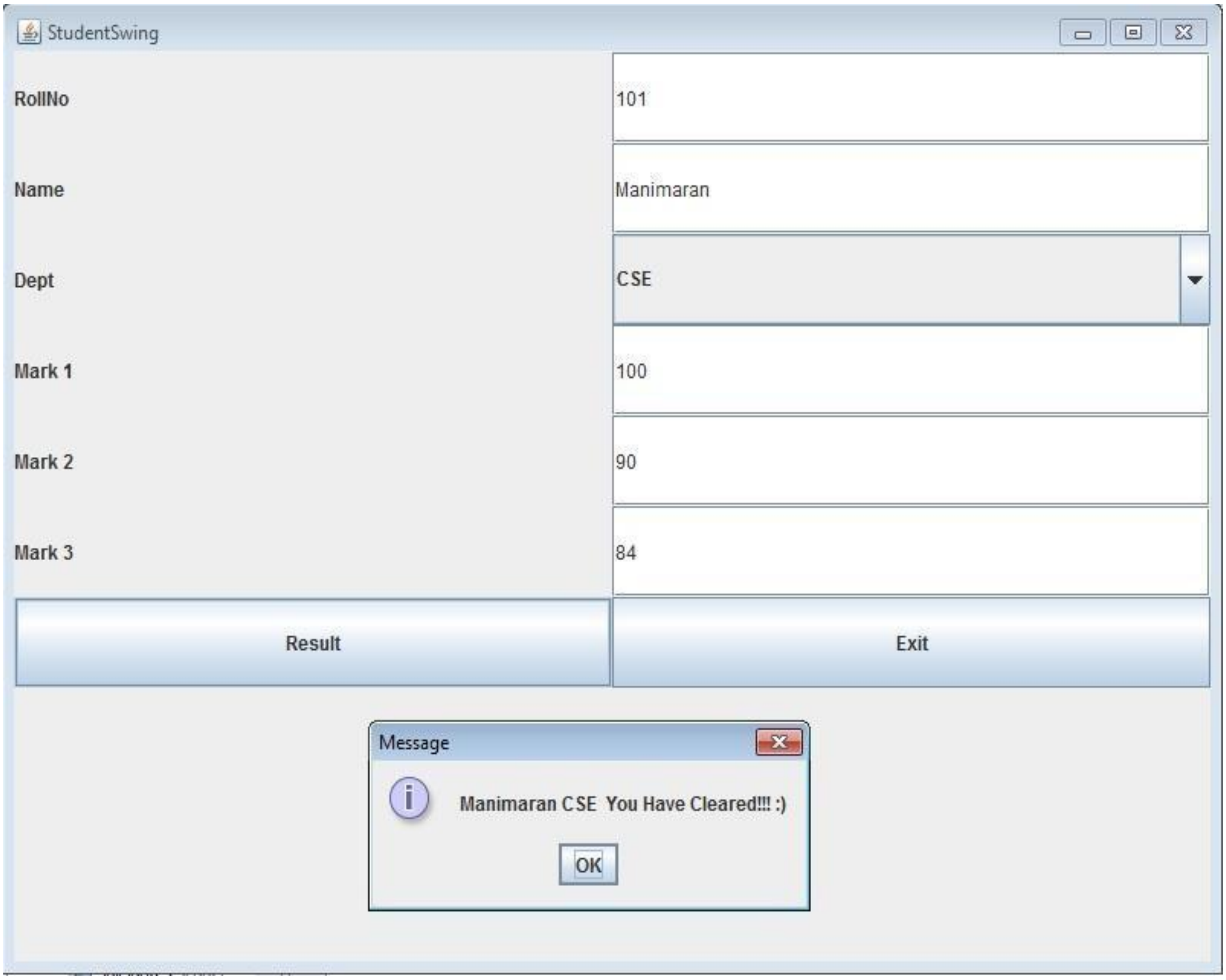
1. Create the class Student Swing which extends the the class J Frame and implements the interface Action Listener.
2. Create a panel and add the label, text fields, combo boxes and buttons to it.
3. Obtain the input for roll number, name, department, and marks from the user.
4. Check if the marks are greater than 50 in the calculate()method.
5. If it is true, print "You have Cleared" else print "Better Luck next time".
6. Create an object for the class and call the methods in the main()function.
7. Finally compile and run the program to verify the result

CODING:

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


public class StudentSwing extends JFrame implements ActionListener
{
  JLabel lblRollno,lblName,lblDept,lblMark1,lblMark2,lblMark3;
  JComboBox comboDept;
  JTextField txtRollno,txtName,txtDept,txtMark1,txtMark2,txtMark3;
  JButton result,exit;

  StudentSwing()
  {
          super("StudentSwing");

          JPanel panel = new JPanel(new GridLayout(10, 2));

          lblRollno= new JLabel("RollNo");
          panel.add(lblRollno);
          txtRollno= new JTextField();
          panel.add(txtRollno);

          lblName= newJLabel("Name");
          panel.add(lblName);
          txtName= newJTextField();
          panel.add(txtName);

          lblDept= new JLabel("Dept");
```

```
        panel.add(lblDept);

        String[] dept={"CSE","IT"};
        comboDept= new JComboBox(dept);
        comboDept.addItem("ECE");
        comboDept.addItem("EEE");
        panel.add(comboDept);

        lblMark1= new JLabel("Mark 1");
        panel.add(lblMark1);
        txtMark1= newJTextField();
        panel.add(txtMark1);

        lblMark2= new JLabel("Mark 2");
        panel.add(lblMark2);
        txtMark2= new JTextField();
        panel.add(txtMark2);

        lblMark3= new JLabel("Mark 3");
        panel.add(lblMark3);
        txtMark3= newJTextField();
        panel.add(txtMark3);

        result = new JButton("Result");
        panel.add(result);
        result.addActionListener(this);

        exit = new JButton("Exit");
        panel.add(exit);
        exit.addActionListener(this);

        add(panel);
        setSize(800, 600);
        this.setVisible(true);

    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==result)
        {
            Calculate();

        }
        else if(e.getSource()==exit)
        {
            this.dispose();
            }
    }

    public void Calculate()
    {
            try
            {
                    int rollno = Integer.parseInt(txtRollno.getText());
                    String name = txtName.getText();
                    String department =comboDept.getSelectedItem().toString();
                    int mark1 =Integer.parseInt(txtMark1.getText());
                    int mark2 =Integer.parseInt(txtMark2.getText());
                    int mark3 =Integer.parseInt(txtMark3.getText());


            if(mark1 >=50 && mark2 >=50 && mark3 >=50)
            {
               JOptionPane.showMessageDialog(this, name +" " + department + " You
                   HaveCleared!!! :) ")
           }
            else
            {
               JOptionPane.showMessageDialog(this, "Dont Worry " + name + " Better Luck Next
              Time :(");
            }

            }
            catch (Exception ex)
            {
                    JOptionPane.showMessageDialog(this, ex);
            }
    }


    public static void main(String[] args)
    {
            StudentSwing ss = new StudentSwing();
            ss.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
  }
```

INPUT & OUTPUT:



StudentSwing

RollNo 101

Name Manimaran

Dept CSE

Mark 1 100

Mark 2 90

Mark 3 84

Message

Manimaran CSE You Have Cleared!!! :)

RESULT:

Thus the swing application program executed and verified successfully

Ex. No: 13	**APPLET**	**Date:**

AIM:

To write a java program to design and implement applet.

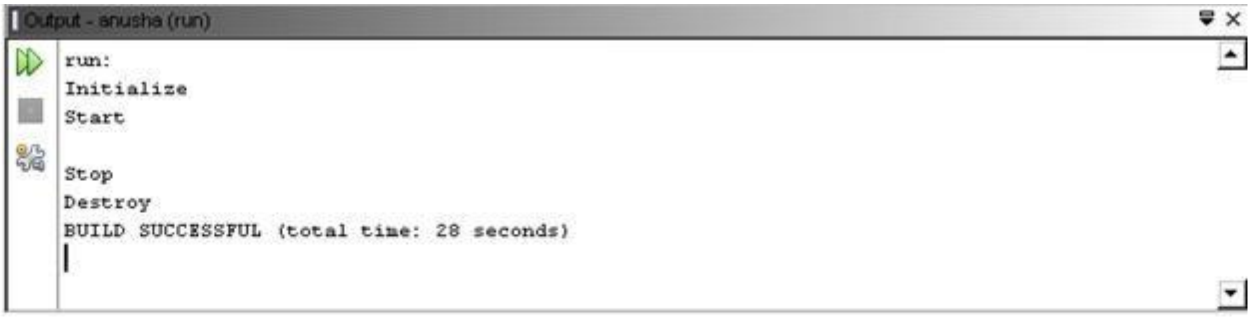
ALGORITHM:

1. Create a class Shape which extends the classApplet.
2. Initialize the thread using init() method and start it using start()method.
3. Draw different shapes such as oval, line and rectangle in the paint()method.
4. The stop() method is used to stop thethread
5. The destroy() method is used to destroy thethread.
6. Compile and run the program to verify theresult

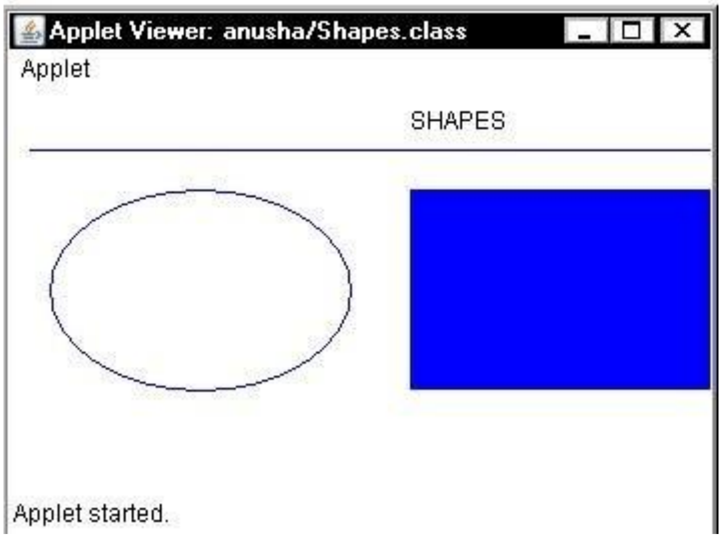
CODING:

```
import java.awt.*;
import java.applet.*;
/*
<applet code="Shapes" width=450 height=250>
</applet>
*/

public class Shapes extends Applet
{
        public void init()
        {
                System.out.println("Initialize");
        }

        public void start()
        {
                System.out.println("Start");
        }

        public void paint(Graphics g)
        {
                g.drawString("SHAPES",200,20);
                g.setColor(Color.blue);
                g.drawLine(10,30,400,30);
                g.drawOval(20,50,150,100);
                g.fillRect(200,50,200,100);
        }

        public void stop()
        {
                System.out.println("Stop");
        }

        public void destroy()
        {
                System.out.println("Destroy");
        }
}
```

INPUT & OUTPUT:



```
run:
Initialize
Start

Stop
Destroy
BUILD SUCCESSFUL (total time: 28 seconds)
```



RESULT:

Thus the applet program executed and verified successfully

Ex. No: 14	JDBC	**Date:**

AIM:

To write a java program to design and implement JDBC

ALGORITHM:

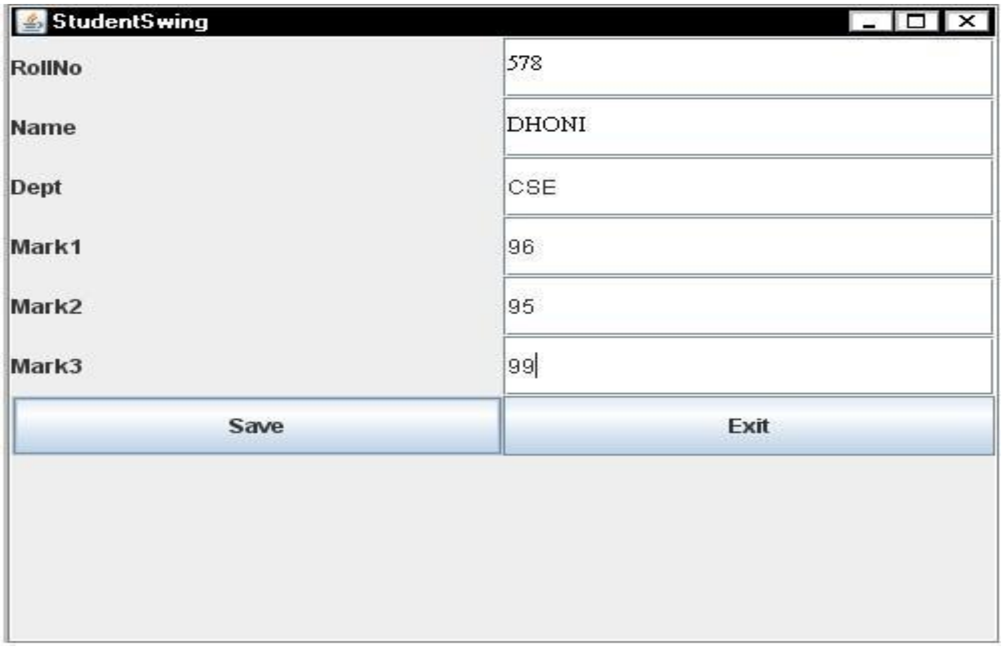
1. Create a class Student Swing Jdbc which extends the class JFrame and implements the Action Listener.
2. Create the panel using JPanel.
3. Use textboxes and labels for rollno, name, department, mark1, mark2,mark3.
4. Add the buttons „Save‟ and„Exit‟.
5. Use the JDBC database connection in the AddStudent() method to add the new student details by using insert command.
6. Check if the marks are greater than 50 and if it is true, print the result as “PASS” and if it is false, print the result as “FAIL”.
7. Create the object for the class in the main()method.
8. Now compile and run the program to see the result.

CODING:

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

public class Student SwingJDBC extends JFrame implements Action Listener
{
  JLabel lblRollno,lblName,lblDept,lblMark1,lblMark2,lblMark3;
  JTextField txtRollno,txtName,txtDept,txtMark1,txtMark2,txtMark3;
  JButton save,exit;

  StudentSwingJDBC()
  {
          super("StudentSwingJDBC");
          JPanel panel = new JPanel(new GridLayout(10,2));
          lblRollno= new JLabel("RollNo");
          panel.add(lblRollno);
          txtRollno= new JTextField();
          panel.add(txtRollno);

          lblName= newJLabel("Name");
          panel.add(lblName);
          txtName= newJTextField();
          panel.add(txtName);
```

```
    lblDept= newJLabel("Dept");
    panel.add(lblDept);
    txtDept= newJTextField();
    panel.add(txtDept);

    lblMark1= new JLabel("Mark 1");
    panel.add(lblMark1);
    txtMark1= newJTextField();
    panel.add(txtMark1);

    lblMark2= new JLabel("Mark 2");
    panel.add(lblMark2);
    txtMark2= newJTextField();
    panel.add(txtMark2);

    lblMark3= new JLabel("Mark 3");
    panel.add(lblMark3);
    txtMark3= newJTextField();
    panel.add(txtMark3);

    save = new JButton("Save");
    panel.add(save);
    save.addActionListener(this);

    exit = new JButton("Exit");
    panel.add(exit);
    exit.addActionListener(this);

    add(panel);
    setSize(500, 400);
    this.setVisible(true);

}

public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==save)
    {
      Add Student();
      JOptionPane.showMessageDialog(this,"Student informationupdated successfully");
    }
    else if(e.getSource()==exit)
    {
        this.dispose();
    }
}
```

```
  public void AddStudent(
  {
    try
    {
       Class.forName("oracle.jdbc.driver.OracleDriver");
      Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE",
            "system", "user");
      int rollno = Integer.parseInt(txtRollno.getText());
       String name = txtName.getText();
       String department = txtDept.getText();
       int  mark1  =Integer.parseInt(txtMark1.getText());
       int  mark2  =Integer.parseInt(txtMark2.getText());
       int  mark3  =Integer.parseInt(txtMark3.getText());
       int total = mark1 + mark2 +mark3;
      String result;
      if(mark1 >=50 && mark2 >=50 && mark3 >=50)
      { result = "Pass";}
      else
      { result = "Fail"; }
       PreparedStatement ps=con.prepareStatement("INSERT INTO mec.mitstudent (rollno,
      name,    department,mark1,mark2,mark3,total,result) VALUES (?,?,?,?,?,?,?,?)");
      ps.setInt(1,rollno);
      ps.setString(2,name);
      ps.setString(3,department);
      ps.setInt(4,mark1);
      ps.setInt(5,mark2);
      ps.setInt(6,mark3);
      ps.setInt(7,total);
      ps.setString(8,result);
      ps.executeQuery();
      ps.close();
      con.close();
      }
          catch (Exception ex)
          {
                  JOptionPane.showMessageDialog(this, ex);
          }
  }

  public static void main(String[] args)
  {
          StudentSwingJDBC ss = new StudentSwingJDBC();
          ss.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
}
```



RollNo 578

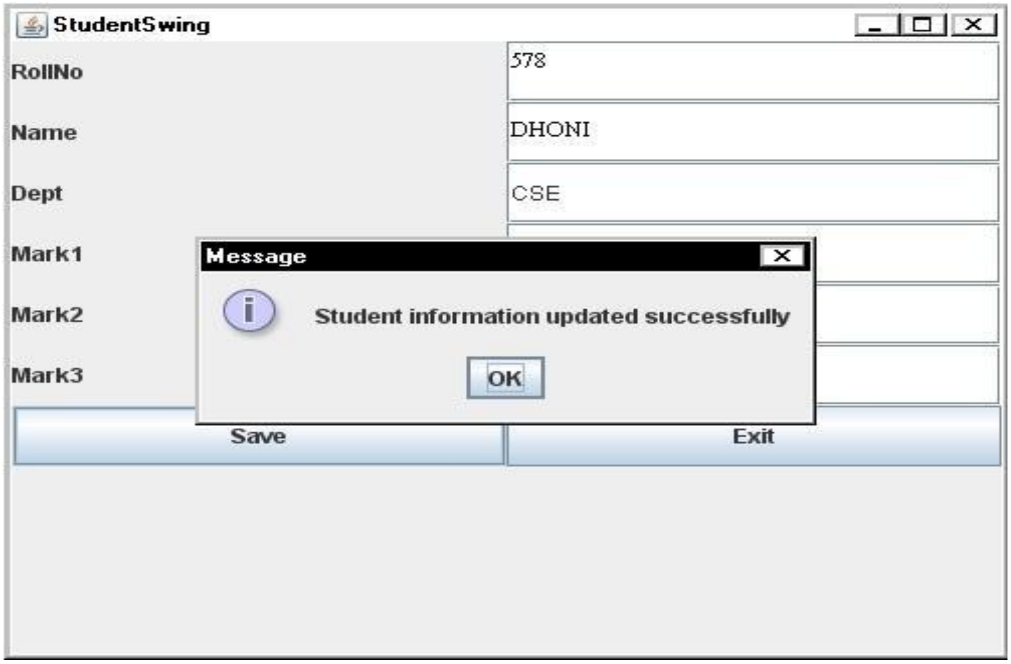
Name DHONI

Dept CSE

Mark1 96

Mark2 95

Mark3 99



RollNo 578

Name DHONI

Dept CSE

Mark1

Mark2

Mark3

Student information updated successfully

RESULT:

Thus the java database connectivity program executed and verified successfully

Ex. No: 15	**EVENT HANDLING- CALCULATOR**	**Date:**

AIM:

To write a java program to implement event handling event for simulating a simple calculator.

ALGORITHM:

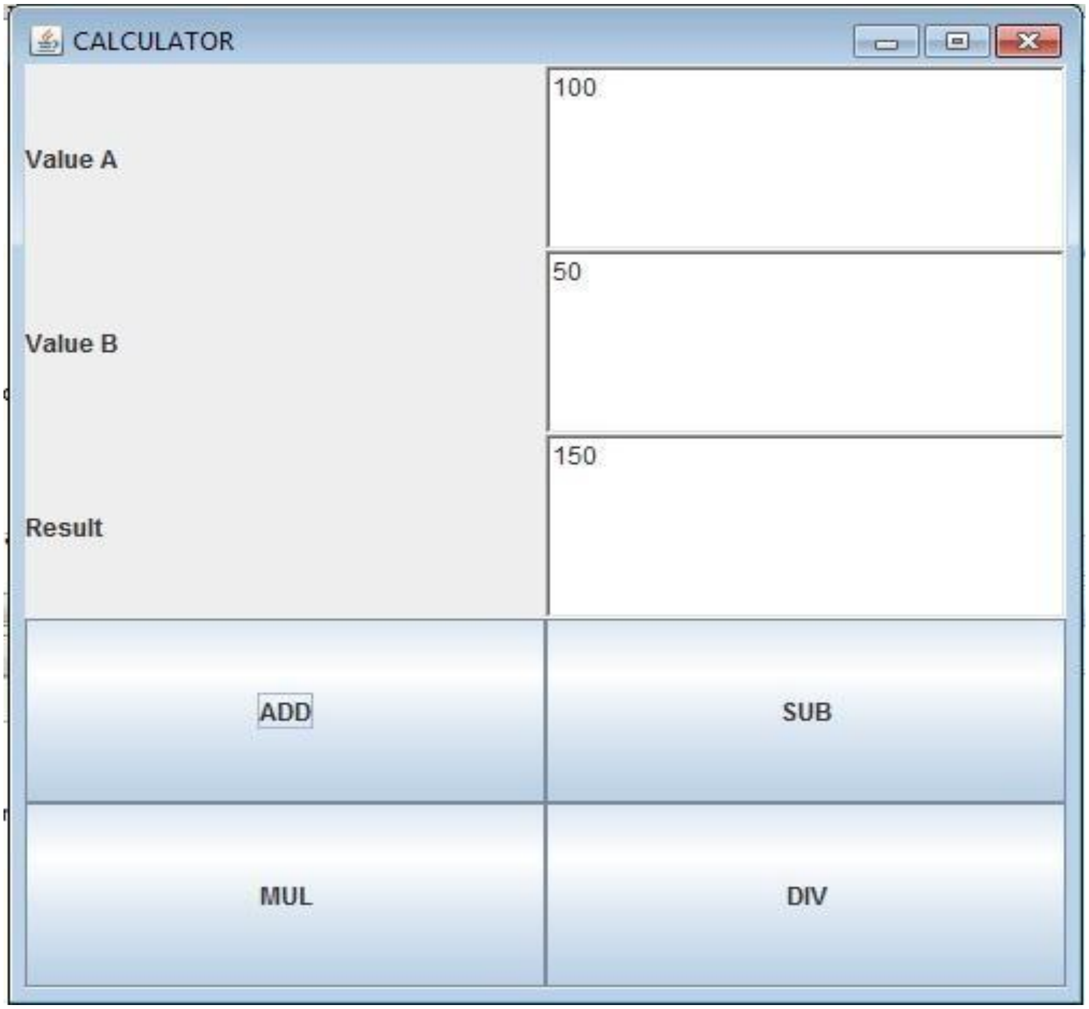
1. Create the calculatorclass
2. Create a frame using JFrame class and a panel using JPanelclass
3. Create the labels ,text fields and buttons to thepanel
4. Create the class Action that extends the class WindowAdapter and implementsthe interface ActionListener andTextListener.
5. Perform the addition operation in actionPerformed() method by overridingit.
6. Set the value for the second textbox if not entered in the textValueChanged()method.
7. Create the object in the main() method and call the othermethods.
8. Finally compile and run theprogram

CODING:

```
package oopl;

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



public class Calculator
{
  TextField t1,t2,t3;
  JButton b1,b2,b3,b4;
  JLabel l1,l2,l3;
  publicCalculator()
  {
    JFrame f=new JFrame("CALCULATOR");
    f.setVisible(true);
    f.setSize(500,500);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    JPanel p=new JPanel();

    l1=new JLabel("Value A");
    l2=new JLabel("Value B");
    l3=new JLabel("Result");

    t1 = new TextField(10);
    t2=new TextField(10);
    t3=new TextField(10);

    b1=new JButton("ADD");
    b2=new  JButton("SUB");
    b3=new JButton("MUL");
    b4=new JButton("DIV");

    p.add(l1);
    p.add(t1);
    p.add(l2);
    p.add(t2);
    p.add(l3);
    p.add(t3);
    p.add(b1);
    p.add(b2);
    p.add(b3);
    p.add(b4);
    p.setLayout(new GridLayout(5,2));
    f.setContentPane(p);
    b1.addActionListener(new Action());
    b2.addActionListener(new Action());
    b3.addActionListener(new Action());
    b4.addActionListener(new Action());
    f.addWindowListener(new Action());

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

}

private class Action extends WindowAdapter implements ActionListener
{

    @Override
    public void actionPerformed(ActionEvent ae)
```

```
        {
            int a=Integer.parseInt(t1.getText());
            int b=Integer.parseInt(t2.getText());

            if(ae.getSource()==b1)
            {
                int c= a+b;
                t3.setText(""+c);
            }

            if(ae.getSource()==b2)
            {
                int c= a-b;
                t3.setText(""+c);
            }

            if(ae.getSource()==b3)
            {
                int c= a*b;
                t3.setText(""+c);
            }

            if(ae.getSource()==b4)
            {
                float c= a/b;
                t3.setText(""+c);
            }
        }

        @Override
        public void windowClosing(WindowEvent we)
        {
            System.exit(0);
        }



    }

}
```

INPUT & OUTPUT:



Value A 100

Value B 50

Result 150

RESULT:

Thus the simulation of calculator program executed and verified successfully

ADDITIONAL PROGRAMS

Ex. No: 16	CURRENCY CONVERTER, DISTANCE CONVERTER AND TIME CONVERTER USING PACKAGES	**Date:**

AIM:

To develop a java application to implement currency converter, distance converter and time converter using packages.

ALGORITHM:

1. The package keyword is used to create a package in java.

2. Create a class Currency Converter inside a package name Currency Converter.

3. Class also contains methods dollor to inr, inr to dollor, euro to inr, inr to euro, yen to inr, and inr to yen with its parameters to convert given currency.

4. Create a class Distance Converter inside a package name Distance Converter.

5. Class also contains methods meter to km, km to meter, miles to km and km to miles with its parameters to convert given distance.

6. Create a class Time Converter inside a package name Time Converter.

7. Class also contains methods hours to minutes, minutes to hours, hours to seconds and seconds to hours with its parameters to convert given time.

8. Import the Currency Converter, Distance Converter, Time Converter and other java packages.

9. Create a class Converter and object for a class in memory and assign it to the reference variable, then the method is invoked. 10. By using Scanner class get the choices for switch statement during runtime. 11. By using switch case statement we can convert currency, distance and time for each choice. 12. Create object for a class in memory and assign it to the reference variable, then the method is invoked. 13. Finally, the conversion is displayed based on type of converter.

PROGRAM:

```
 //For Packages, Folder Name should be CurrencyConverter
//File Name should be CurrencyConverter.java package CurrencyConverter;
 public class CurrencyConverter
{
 public double dollortoinr(double x)
 {
double inr=x*67.86; return inr;
 }
public double inrtodollor(double x)
{
double dollor=x/67.86; return dollor;
 }
public double eurotoinr(double x)
```

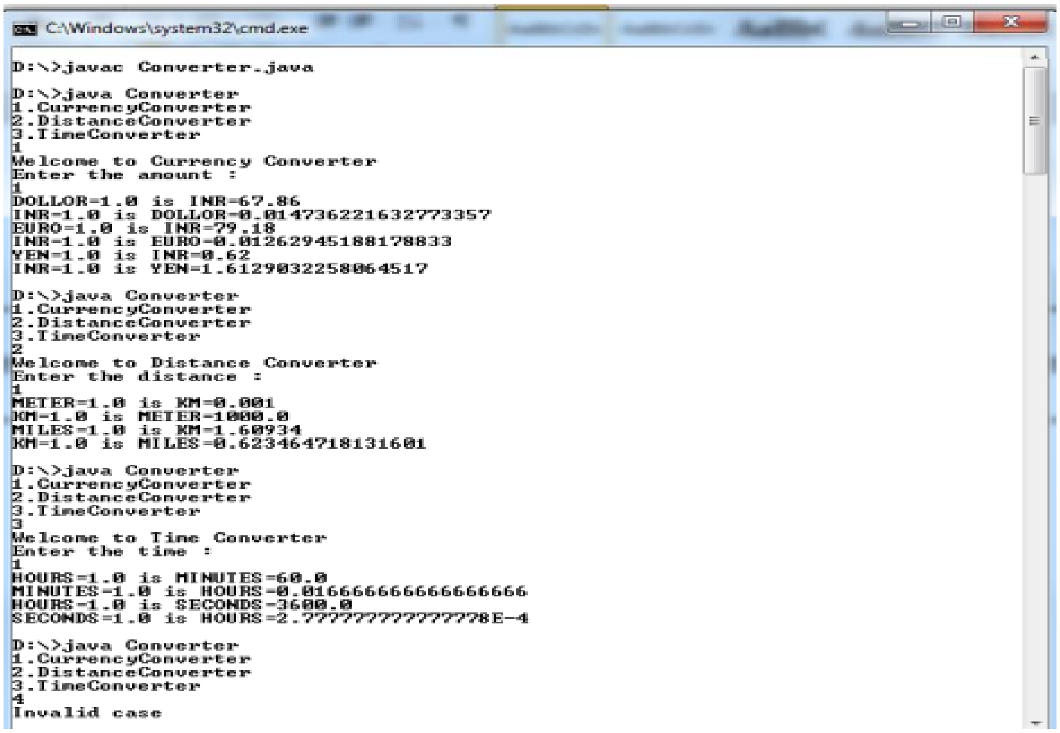
```
 {
double inr=x*79.18; return inr;
}
public double inrtoeuro(double x)
 {
double euro=x/79.18; return euro;
 }
public double yentoinr(double x)
 {
double inr=x*0.62; return inr;
}
public double inrtoyen(double x)
 {
double yen=x/0.62; return yen;
 } }
//For Packages, Folder Name should be DistanceConverter
//File Name should be DistanceConverter.java package DistanceConverter;
 public class DistanceConverter
{
public double metertokm(double x)
 {
double km=x*0.001; return km;
}
public double kmtometer(double x)
 {
double meter=x/0.001;
 return meter;
}
public double milestokm(double x)
 {
double km=x*1.60934;
 return km;
 }
public double kmtomiles(double x)
{
double miles=x/1.60394; return miles;
```

```
 } }
 //For Packages, Folder Name should be Time Converter
//File Name should be TimeConverter.java package Time Converter;
 public class Time Converter
 {
 public double hours to minutes(double x)
{
double minutes=x*60;
return minutes;
}
public double minutes to hours(double x)
{
double hours=x/60;
 return hours;
}
 public double hours to seconds(double x)
 {
 double seconds=x*3600;
return seconds;
}
public double seconds to hours(double x)
 {
double hours=x/3600; return hours;
} }
//File Name should be Converter.java separate this file from above 3 folders import CurrencyConverter.*;
import DistanceConverter.*;
 import TimeConverter.*;
import java.io.*;
import java.util.*;
 class Converter
 {
 public static void main(String args[])
{
System.out.println("1.CurrencyConverter");
System.out.println("2.DistanceConverter");
System.out.println("3.TimeConverter");
```

```
Converter cr = new Converter();
Scanner c = new Scanner(System.in);
int choice = c.nextInt();
String op = null; switch(choice)
 {
 case 1: cr.Currency();
 break;
case 2: cr.Distance();
break;
case 3: cr.Time();
break;
default:
System.out.println("Invalid case");
return;
} }
public void Currency()
{
 Scanner in = new Scanner(System.in);
System.out.println("Welcome to Currency Converter");
System.out.println("Enter the amount :");
double amt = in.nextInt(); CurrencyConverter cc = new CurrencyConverter();
System.out.println("DOLLOR="+amt+" is INR="+cc.dollortoinr(amt));
System.out.println("INR="+amt+" is DOLLOR="+cc.inrtodollor(amt));
System.out.println("EURO="+amt+" is INR="+cc.eurotoinr(amt));
 System.out.println("INR="+amt+" is EURO="+cc.inrtoeuro(amt));
System.out.println("YEN="+amt+" is INR="+cc.yentoinr(amt));
 System.out.println("INR="+amt+" is YEN="+cc.inrtoyen(amt));
 }
 public void Distance()
 {
 Scanner in = new Scanner(System.in);
System.out.println("Welcome to Distance Converter");
System.out.println("Enter the distance :");
double dis = in.nextInt();
```

```
DistanceConverter dd = new DistanceConverter();
System.out.println("METER="+dis+" is KM="+dd.metertokm(dis));
System.out.println("KM="+dis+" is METER="+dd.kmtometer(dis));
System.out.println("MILES="+dis+" is KM="+dd.milestokm(dis));
System.out.println("KM="+dis+" is MILES="+dd.kmtomiles(dis));
 }
 public void Time()
 {
Scanner out = new Scanner(System.in);
System.out.println("Welcome to Time Converter");


 System.out.println("Enter the time :");
 double tim = out.nextInt();
 TimeConverter tt = new TimeConverter();

 System.out.println("HOURS="+tim+" is MINUTES="+tt. hourstominutes(tim));
System.out.println("MINUTES="+tim+" is HOURS="+tt.minutestohours(tim));
System.out.println("HOURS="+tim+" is SECONDS="+tt.hourstoseconds(tim));
System.out.println("SECONDS="+tim+" is HOURS="+tt.secondstohours(tim));
}
 }
```

OUTPUT:



```
D:\>javac Converter.java

D:\>java Converter
1.CurrencyConverter
2.DistanceConverter
3.TimeConverter
1
Welcome to Currency Converter
Enter the amount :
1
DOLLOR=1.0 is INR=67.86
INR=1.0 is DOLLOR=0.014736221632773357
EURO=1.0 is INR=79.18
INR=1.0 is EURO=0.012629451881788833
YEN=1.0 is INR=0.62
INR=1.0 is YEN=1.6129032258064517

D:\>java Converter
1.CurrencyConverter
2.DistanceConverter
3.TimeConverter
2
Welcome to Distance Converter
Enter the distance :
1
METER=1.0 is KM=0.001
KM=1.0 is METER=1000.0
MILES=1.0 is KM=1.60934
KM=1.0 is MILES=0.6213464718131601

D:\>java Converter
1.CurrencyConverter
2.DistanceConverter
3.TimeConverter
3
Welcome to Time Converter
Enter the time :
1
HOURS=1.0 is MINUTES=60.0
MINUTES=1.0 is HOURS=0.016666666666666666
HOURS=1.0 is SECONDS=3600.0
SECONDS=1.0 is HOURS=2.777777777777778E-4

D:\>java Converter
1.CurrencyConverter
2.DistanceConverter
3.TimeConverter
4
Invalid case
```

RESULT:

Thus the application for currency converter, distance converter and time converter using packages has been successfully executed.