DEPARTMENT OF COMPUTER APPLICATION

TKM COLLEGE OF ENGINEERING

KOLLAM – 691005

20MCA132 -OBJECT ORIENTED PROGRAMMING LAB

PRACTICAL RECORD BOOK

Second Semester MCA

2021-2022

Submitted by:

NAME: SONI R

ROLL NO: TKM21MCA-2036

DEPARTMENT OF COMPUTER APPLICATION

TKM COLLEGE OF ENGINEERING

KOLLAM – 691005

Certificate

This is a bonafide record of the work done by SONI R (TKM21MCA-2036) in the Second Semester in OBJECT ORIENTED PROGRAMMING LAB Course(20MCA132) towards the partial fulfillment of the degree of Master of Computer Applications during the academic year 2021-2022.

Staff Member in-charge

……..

Examiner

……..

COURSE OUTCOME-1

PROGRAM-1

AIM: Define a class ‘product’ with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

PROGRAM CODE

```
package CO1;
import java.util.*;
class p {
   String pcode, pname;
   float price;
   Scanner s = new Scanner(System.in);
   p() {
      System.out.println("Enter Product name");
      pname = s.nextLine();
      System.out.println("Enter Product code");
      pcode = s.nextLine();
      System.out.println("Enter Product price");
      price = s.nextFloat();
   }
}
public class pg_1 {
   public static void main(String[] args) {
      int i = 0;
      p o[] = new p[3];
      while (i < 3) {
         p obj = new p();
         o[i] = obj;
         i++;
      }
      if (o[0].price < o[1].price && o[0].price < o[2].price) {
         System.out.println("Product " + o[0].pname + " has lowest price
: " + o[0].price);
      } else if (o[1].price < o[0].price && o[1].price < o[2].price) {
         System.out.println("Product " + o[1].pname + " has lowest price
: " + o[1].price);
      } else {
         System.out.println("Product " + o[2].pname + " has lowest price
: " + o[2].price);
      }
   }
}
```

OUTPUT:

```
Enter Product name
cake
Enter Product code
101
Enter Product price
100
Enter Product name
sweets
Enter Product code
102
Enter Product price
120
Enter Product name
bread
Enter Product code
152
Enter Product price
50
Product bread has lowest price  : 50.0
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-2

AIM: Read 2 matrices from the console and perform matrix addition.

PROGRAM CODE

```
package CO1;
import java.util.*;
class mat {
   int a[][] = new int[10][10];
   int r, c;
   Scanner s = new Scanner(System.in);
   int[][] get() {
      System.out.println("Enter number of columns");
      r = s.nextInt();
      System.out.println("Enter number of rows");
      c = s.nextInt();
      System.out.println("Enter number of rows");
      for (int i = 0; i < r; i++) {
         for (int j = 0; j < c; j++) {  a[i][j] = s.nextInt();}        }
      return a;    }
   void display(int a[][]) {
      for (int i = 0; i < r; i++) {
         for (int j = 0; j < c; j++) {
            System.out.print(a[i][j]+"\t");
         System.out.println("");        }}
   int[][] add(int a[][], int b[][]) {
      int c1[][] = new int[10][10];
      for (int i = 0; i < r; i++) {
         for (int j = 0; j < c; j++) {
            c1[i][j] = a[i][j] + b[i][j];            }}
      return c1;    }}
public class pg_2 {
   public static void main(String[] args) {
      mat m1 = new mat();
      int a[][] = m1.get();

 mat m2 = new mat();
      int b[][] = m2.get();
      System.out.println("Sum is");
      int c[][] = m1.add(a, b);
      m1.display(c);    }}
```

OUTPUT:

```
Enter number of columns
3
Enter number of rows
3
Enter matrix
1 1 1
1 1 1
1 1 1
Enter number of columns
3
Enter number of rows
3
Enter matrix
2 2 2
2 2 2
2 2 2
Sum is
3       3       3
3       3       3
3       3       3
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-3

AIM : Add complex numbers

PROGRAM CODE

```
package CO1;
import java.util.Scanner;

class complex {
   void add(String s1, String s2) {
      int a1 = Integer.parseInt((s1.split("\\+"))[0]);
      int b1 = Integer.parseInt((s1.split("\\+")[1]).split("i")[0]);
      int a2 = Integer.parseInt((s2.split("\\+"))[0]);
      int b2 = Integer.parseInt((s2.split("\\+")[1]).split("i")[0]);
      System.out.println("Sum is = " + (a1 + a2) + "+" + (b1 + b2 +
"i"));
   }
}
public class pg_3 {
   public static void main(String[] args) {
      Scanner s = new Scanner(System.in);
      complex c = new complex();
      System.out.println("Enter a complex number");
      String s1 = s.nextLine();
      System.out.println("Enter a complex number");
      String s2 = s.nextLine();
      c.add(s1, s2);
   }
}
```

OUTPUT:

```
Enter a complex number
1+2i
Enter a complex number
2+3i
Sum is = 3+5i
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-4

AIM: Read a matrix from the console and check whether it is symmetric or not.

PROGRAM CODE

```
package CO1;
import java.util.Scanner;
class sym {
    int a[][] = new int[10][10];
    int r, c;
    Scanner s = new Scanner(System.in);
    int[][] get() {
        System.out.println("Enter number of columns");
        r = s.nextInt();
        System.out.println("Enter number of rows");
        c = s.nextInt();
        System.out.println("Enter the" + r + " x " + c + "matrix");
        for (int i = 0; i < r; i++)
            for (int j = 0; j < c; j++)
                a[i][j] = s.nextInt();
        return a;    }
    void CheckSym(int a[][]) {
        int d = 0;
        for (int i = 0; i < r; i++)
            for (int j = 0; j < c; j++)
                if (a[i][j] != a[j][i])
                    d = 1;
        if (d == 0)
            System.out.println("Matrix is Symemtric");
        else
            System.out.println("Matrix is not Symemtric");    }}
public class pg_4 {
    public static void main(String[] args) {
        int a[][] = new int[10][10];
        sym s1 = new sym();
        a = s1.get();
        //s1.display(a);
        s1.CheckSym(a);
    }}
```

OUTPUT:

```
Enter number of columns
3
Enter number of rows
3
Enter the3 x 3matrix
1 1 1
1 1 1
1 1 1
Matrix is Symemtric
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-5

AIM: Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

PROGRAM CODE

```
package CO1;
 public class pg_5 {
    public static void main(String[] args) {
        cpu c = new cpu();
        cpu.processor p = c.new processor(2, "AMD");
        cpu.ram r = new cpu.ram(4, "Crucial");
        p.display();
        r.display();
        System.out.println();    }}


class cpu {
    int price = 0;
    //nested class
    class processor {
        int cores;
        String manufacturer;
        processor(int x, String s) {
            cores = x;
            manufacturer = s;
        }
        void display() {
            System.out.println("Processor - \nCores - " + cores + "\n" +
"Manufacturer - " + manufacturer+"\n");
        }
    }

    static class ram {
        int mem;
        String manufacturer;
        ram(int x, String s) {
            mem = x;
            manufacturer = s;
        }
void display() {
            System.out.println("Ram - \nMemory - " + mem + "\n" +
"Ram Manufacturer - " + manufacturer+"\n");
        }
    }
}
```

OUTPUT:

```
Processor -
Cores - 2
Manufacturer - AMD

Ram -
Memory - 4
Manufacturer - Crucial
```

RESULT:

The above program is successfully executed and obtained the output

COURSE OUTCOME – 2

PROGRAM-6

AIM: Program to Sort strings

PROGRAM CODE	package CO2;
	import java.util.*;
	public class pg_1 {
	  public static void main(String[] args) {
	  String k, t;
	  String a[] = new String[10];
	  Scanner s = new Scanner(System.in);
	  System.out.println("Enter a sentence");
	  k = s.nextLine();
	  a = k.split(" ");
	  for (int i = 0; i < a.length; i++)
	  for (int j = i + 1; j < a.length; j++)
	  if (a[i].compareTo(a[j]) > 0) {
	  t = a[i];
	  a[i] = a[j];
	  a[j] = t;
	  }
	  System.out.println("Sorted Order : ");
	  for (int i = 0; i < a.length; i++)
	  System.out.println(a[i]);
	  }
	}

OUTPUT:

```
Enter a sentence
what is this thing
Sorted Order :
is
thing
this
what
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-7

AIM: Search an element in an array.

PROGRAM CODE

```
package CO2;
import java.util.*;
public class pg_2 {
    public static void main(String[] args) {
        int[] a = new int[10];
        int b = 0, k = 0;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter number of elements");
        k = s.nextInt();
        System.out.println("Enter the elements");
        for (int i = 0; i < k; i++) {
            a[i] = s.nextInt();
        }
        System.out.println("Enter the number to search");
        int x = s.nextInt();
        for (int i = 0; i < k; i++) {
            if (a[i] == x) {
                System.out.println("Number found");
                b = 1;
                break;
            }
        }
        if (b == 0)
            System.out.println("Number not found");
    }
}
```

OUTPUT:

```
Enter number of elements
5
Enter the elements
4
8
9
6
7
Enter the number to search
6
Number found
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-8

AIM: Perform string manipulations

PROGRAM CODE	package CO2;
	
	public class pg_3 {
	  public static void main(String[] args) {
	  String s = " What is this place ";
	  System.out.println("UpperCase : " + s.toUpperCase());
	  System.out.println("LowerCase : " + s.toLowerCase());
	  System.out.println("Trim : " + s.trim());
	  System.out.println("Replace : " + s.replace("place", "forest"));
	  System.out.println("Length : " + s.length());
	  System.out.println("Character at position : " + s.charAt(4));
	  }
	}
	
	}

OUTPUT:

```
UpperCase :     WHAT IS THIS PLACE
LowerCase :     what is this place
Trim : What is this place
Replace :     What is this forest
Length : 26
Character at position : W
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-9

AIM: Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, usingthe concept of Array of Objects.

PROGRAM CODE

```
package CO2;
import java.util.Scanner;
class employee {
    int eNo;
    String eName;
    Double eSalary;
    employee(int a, String b, Double c) {
        eNo = a;
        eName = b;
        eSalary = c;    }}
public class pg_4 {
    public static void main(String[] args) {
        int i = 0, f = 0;
        employee o[] = new employee[10];
        System.out.println("Enter the number of employees");
        Scanner s = new Scanner(System.in);
        int n = s.nextInt();
        while (i < n) {
            System.out.println("Enter employee id");
            int a = s.nextInt();
            System.out.println("Enter employee name");
            s.nextLine();
            String b = s.nextLine();
            System.out.println("Enter employee salary");
            Double c = s.nextDouble();
            o[i] = new employee(a, b, c);
                                    i++;        }
System.out.println("Enter employee id to search details");
        int a = s.nextInt();
        i = 0;
        while (i < n) {
            if (o[i].eNo == a) {
                System.out.println("Employee Found ");
                System.out.println("eNo : " + o[i].eNo + "\neName : " +
o[i].eName + "\neSalary : " + o[i].eSalary);
                f = 1;            }
            i++;        }
        if (f == 0) {            System.out.println("Employee Not Found");
}    }}
```

OUTPUT:

```
Enter the number of employees
2
Enter employee id
101
Enter employee name
Rob
Enter employee salary
15000
Enter employee id
102
Enter employee name
Tom
Enter employee salary
45000
Enter employee id to search details
102
Employee Found
eNo : 102
eName : Tom
eSalary : 45000.0
```

RESULT:

The above program is successfully executed and obtained the output

COURSE OUTCOME-3

PROGRAM-10

AIM: Area of different shapes using overloaded functions

PROGRAM CODE

```
import java.util.Scanner;
public class overload {
	void area(int a){
		System.out.println("Area of Square= "+a*a);
	}
	void area(int l,int b){
		System.out.println("Area of Rectangle= "+l*b);
	}
	void area(float r){
		System.out.println("Area of Circle= "+3.14*r*r);
	}

	public static void main(String[] args) {
		int ch;
		Scanner s =new Scanner(System.in);
		overload fn=new overload();
		System.out.println("1-Square\n2-Rectangle\n3-
Circle\n0-Exit");
		do{
			System.out.println("Enter the choice");
			ch=s.nextInt();
			switch(ch){
	case 1:System.out.println("Enter the side of the square");
					int x=s.nextInt();
					fn.area(x);
					break;
				case 2:System.out.println("Enter the length and
breadth of the rectangle");
					int y=s.nextInt();
					int z=s.nextInt();
					fn.area(y,z);
					break;
				case 3:System.out.println("Enter the radius of
the circle ");
					float r=s.nextFloat();
					fn.area(r);
					break;}
		}while(ch!=0);
		s.close();
	}

 }
```

OUTPUT:

```
1-Square
2-Rectangle
3-Circle
0-Exit
Enter the choice
1
Enter the side of the square
5
Area of Square= 25
Enter the choice
2
Enter the length and breadth of the rectangle
2
3
Area of Rectangle= 6
Enter the choice
3
Enter the radius of the circle
10
Area of Circle= 314.0
Enter the choice
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-11

AIM: Create a class ‘Employee’ with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class ‘Teacher’ that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers

PROGRAM CODE

```
import java.util.Scanner;
public class myclass {
	public static void main(String[] args) {
		Scanner s=new Scanner(System.in);
		System.out.println("Enter the number of employees");
		int n=s.nextInt();
		s.nextLine();
		Teacher obj[]= new Teacher[n];
		for(int i=0;i<n;i++){
			System.out.println("Teacher "+(i+1));
			System.out.println("Enter the ID");
			int u=s.nextInt();
			s.nextLine();
			System.out.println("Enter the name");
			String v=s.nextLine();
			System.out.println("Enter the Address");
			String w=s.nextLine();
			System.out.println("Enter the Salary");
			int x=s.nextInt();
			s.nextLine();
			System.out.println("Enter the Department");
			String y=s.nextLine();
			System.out.println("Enter the Subject");
			String z=s.nextLine();
			obj[i]=new Teacher(u,v,w,x,y,z);
		}
		System.out.println("DETAILS\n");
		for(int i=0;i<n;i++){
			System.out.println("Teacher "+(i+1));
			obj[i].display();
		}

	}

}

class Employee3 {
	int id,salary;
	String name,address;
	Employee3(int a,String b,String c,int d){
```

```
                        this.id=a;
                        this.name=b;
                        this.salary=d;
                        this.address=c;
        }
}

class Teacher extends Employee3{
        String dept,sub;
        Teacher(int a,String b,String c,int d,String e,String f){
                super(a,b,c,d);
                this.dept=e;
                this.sub=f;
        }
        void display(){
                System.out.println("ID"+id+"\nName: "+name);
                System.out.println("Address: "+address);
                System.out.println("Salary: "+salary);
                System.out.println("Department: "+dept);
                System.out.println("Subject: "+sub);
        }

}
```

OUTPUT:

```
DETAILS

Teacher 1
ID101
Name: Binoy
Address: BinoyHome
Salary: 100000
Department: CS
Subject: Cpp
Teacher 2
ID102
Name: Kannan
Address: Kannanvilla
Salary: 100001
Department: CS
Subject: Dbms
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-12

AIM: Create a class 'Person' with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class 'Employee' thatinherits the properties of class Person and also contains its own data members like Empid, Company_name, Qualification, Salary and its own constructor. Create another class 'Teacher' that inherits the properties of class Employee and containsits own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

PROGRAM CODE

```
import java.util.Scanner;
public class multilevel {
	public static void main(String[] args) {
		Scanner s=new Scanner(System.in);
		System.out.println("Enter the number of employees");
		int n=s.nextInt();	s.nextLine();
		Teacher obj[]= new Teacher[n];
		for(int i=0;i<n;i++){
			System.out.println("Teacher "+(i+1));
			System.out.println("Enter the name");
			String a=s.nextLine();
			System.out.println("Enter the age");
			int b=s.nextInt();	s.nextLine();
			System.out.println("Enter the gender");
			String c=s.nextLine();
			System.out.println("Enter the address");
			String d=s.nextLine();
			System.out.println("Enter the Employee ID");
			int e1=s.nextInt();	s.nextLine();
			System.out.println("Enter the Company name");
			String e2=s.nextLine();
			System.out.println("Enter the Qualification");
			String e3=s.nextLine();
			System.out.println("Enter the salary");
			int e4=s.nextInt();	s.nextLine();
			System.out.println("Enter the Teacher ID");
			int t1=s.nextInt();	s.nextLine();
			System.out.println("Enter the Department");
			String t2=s.nextLine();
			System.out.println("Enter the Subject");
			String t3=s.nextLine();
			obj[i]=new Teacher(a,b,c,d,e1,e2,e3,e4,t1,t2,t3);
		}
		System.out.println("\nDETAILS");
		for(int i=0;i<n;i++){
			System.out.println("\nTeacher"+(i+1));
			obj[i].display();
```

```
                }}
}
class Person{
        int age;
        String name,gender,address;
        Person(String a,int b,String c,String d){
                this.name=a;
                this.age=b;
                this.gender=c;
                this.address=d;
        }
}
class Employeee extends Person{
        int empid,salary;
        String comp_name,qualification;
        Employeee(String a,int b,String c,String d,int eid,String
cname,String quali,int sal){
                        super(a,b,c,d);
                        this.empid=eid;
                        this.comp_name=cname;
                        this.qualification=quali;
                        this.salary=sal;
        }
}
class Teacher extends Employeee{
        String dept,sub;
        int t_id;
        Teacher(String a,int b,String c,String d,int eid,String
cname,String quali,int sal,int tid,String dep,String subj){
                super(a,b,c,d,eid,cname,quali,sal);
                this.dept=dep;
                this.sub=subj;
                this.t_id=tid;
        }
        void display(){

        System.out.println("Name:"+name+"\nAge:"+age+"\nGender:"
+gender+"\nAddress"+address+"\nEmployee
ID:"+empid+"\nCompany
Name:"+comp_name+"Salary:"+salary+"\nQualifications:"+qualificati
on+"\nTeacher ID:"+t_id+"\nDepartment:"+dept+"\nSubject:"+sub);
        }
 }
```

OUTPUT:

```
Enter the number of employees
2
Teacher 1
Enter the name
Binoy
Enter the age
21
Enter the gender
M
Enter the address
Binoyhouse
Enter the Employee ID
101
Enter the Company name
TCS
Enter the Qualification
MCA
Enter the salary
100000
Enter the Teacher ID
201
Enter the Department
Training
Enter the Subject
Cpp
Teacher 2
Enter the name
Kanann
Enter the age
23
Enter the gender
M
Enter the address
Kannanvilla
Enter the Employee ID
102
Enter the Company name
Enter the Qualification
MCA
Enter the salary
100000
Enter the Teacher ID
202
Enter the Department
Testing
Enter the Subject
Alpha Test

DETAILS

Teacher1
Name:Binoy
Age:21
Gender:M
AddressBinoyhouse
Employee ID:101
Company Name:TCSSalary:100000
Qualifications:MCA
Teacher ID:201
Department:Training
Subject:Cpp

Teacher2
Name:Kanann
Age:23
Gender:M
AddressKannanvilla
Employee ID:102
Company Name:InfosysSalary:100000
Qualifications:MCA
Teacher ID:202
Department:Testing
Subject:Alpha Test
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-13

AIM: Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance

PROGRAM CODE

```
import java.util.Scanner;
public class myclass{
	public static void main(String[] args){
	Scanner s=new Scanner(System.in);
		System.out.println("Enter the number of Literature books");
		int l=s.nextInt();
		Literature lbook[]=new Literature[l];
		for(int i=0;i<l;i++){
			System.out.println("Enter the details of Literature book "+(i+1));
			lbook[i]=new Literature();
		}
		System.out.println("Enter the number of Fiction books");
		int f=s.nextInt();
		Fiction fbook[]=new Fiction[f];
		for(int i=0;i<f;i++){
			System.out.println("Enter the details of Fiction book"+(i+1));
			fbook[i]=new Fiction();
		}
		int ch;
		do{
			System.out.println("\n1-Literature Book Details\n2-Fiction\n0-Exit\nEnter the choice");
			ch=s.nextInt();
			switch(ch){
				case 1:

	System.out.println("\n**LITERATURE BOOKS**");
					for(int i=0;i<l;i++){
						lbook[i].display();
					}
					break;
				case 2:
					System.out.println("**FICTION BOOKS**");
					for(int i=0;i<f;i++){
						fbook[i].display();
					}
			}
		}while(ch!=0);}}
class Publisher{
	String pub_name;
```

```
        Scanner s=new Scanner(System.in);
        Publisher(){
                System.out.println("Enter the publisher name");
                pub_name=s.nextLine();
        }
}
class Book extends Publisher{
        String b_name,author;
        Book(){
                System.out.println("Enter the book name");
                b_name=s.nextLine();
                System.out.println("Enter the author name");
                author=s.nextLine();
        }
}
class Literature extends Book{
        String type;
        Literature(){
                type="Literature";

        }
        void display(){
                System.out.println("\nBook
Name:"+b_name+"\nAuthor"+author+"\nPublisher:"+pub_name+"\nT
ype:"+type);
        }
}
class Fiction extends Book{
        String type;
        Fiction(){
                type="Fiction";
        }
        void display(){
                System.out.println("\nBook
Name:"+b_name+"\nAuthor"+author+"\nPublisher:"+pub_name+"\nT
ype:"+type);
        }}
```

OUTPUT:

```
Enter the number of Literature books
2
Enter the details of Literature book 1
Enter the publisher name
WB
Enter the book name
Harrypoter
Enter the author name
J K R
Enter the details of Literature book 2
Enter the publisher name
Fox
Enter the book name
LOTR
Enter the author name
Gandalf
Enter the number of Fiction books
1
Enter the details of Fiction book1
Enter the publisher name
DC
Enter the book name
Batman
Enter the author name
Bale


**LITERATURE BOOKS**

Book Name:Harrypoter
AuthorJ K R
Publisher:WB
Type:Literature

Book Name:LOTR
AuthorGandalf
Publisher:Fox
Type:Literature

1-Literature Book Details
2-Fiction
0-Exit
Enter the choice
2
**FICTION BOOKS**

Book Name:Batman
AuthorBale
Publisher:DC
Type:Fiction
```

RESULT:

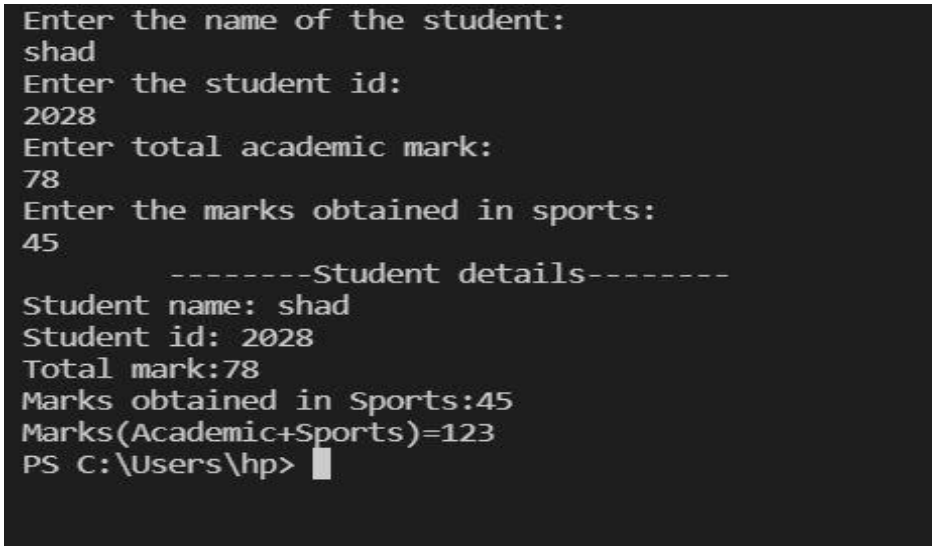
The above program is successfully executed and obtained the output

PROGRAM-14

AIM: Create classes Student and Sports. Create another class Result inheritedfrom Student and Sports. Display the academic and sports score of a student

PROGRAM CODE	import java.util.Scanner;
	class student{ int roll; String name;
	int phy,chem,maths;
	student(){
	Scanner s= new Scanner(System.in); System.out.println("enter the roll num"); roll =s.nextInt(); System.out.println("enter name"); name=s.next();
	System.out.println("enter phyics marks"); phy =s.nextInt();
	System.out.println("enter chemisty marks"); chem =s.nextInt(); System.out.println("enter maths marks"); maths =s.nextInt();
	}
	}
	class sports extends student{
	int score; sports(){
	Scanner sc= new Scanner(System.in); System.out.println("enter sports score"); score=sc.nextInt();
	}
	}
	class result extends sports{
	void diplay(){
	System.out.println("----------Academic Details ");
	System.out.println("Name : " + name);
	System.out.println("Roll No : " + roll); System.out.println(""); System.out.println("----------MARKS ");
	System.out.println("Physics :" + phy); System.out.println("Chemistry :" + chem); System.out.println("Maths :" + maths); System.out.println("");
	System.out.println("----------SPORTS SCORE ");
	System.out.println("Score : " + score);
	System.out.println("");
	}

OUTPUT:



RESULT:

The above program is successfully executed and obtained the output

PROGRAM-15

AIM: Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

PROGRAM CODE

```
import java.util.Scanner;
public class Menu {
	public static void main(String[] args) {
		Scanner s=new Scanner(System.in);
		int ch;
		Inter I;
		Circle obj1= new Circle();

		Rectangle obj2=new Rectangle();
		I=obj2;
		do{
			System.out.println("1-Area of Circle\n2-
Perimeter of Circle\n3-Area of Rectangle\n4-Perimeter of
Rectangle\n0-Exit\nEnter the choice");
			ch=s.nextInt();
			switch(ch){
			case 1:I=obj1;
			I.area();
				break;
			case 2:I=obj1;
			I.perimeter();
			break;
			case 3:I=obj2;
			I.area();
				break;
			case 4:I=obj2;
			I.perimeter();
			break;
			default:System.out.println("Invalid");
			}
		}while(ch!=0);
	}

}
interface Inter{
	void area();
	void perimeter();

}
class Circle implements Inter{
	public void area(){
		Scanner s=new Scanner(System.in);
		System.out.println("Enter the radius");
		int r=s.nextInt();
```

```
                System.out.println("Area of Circle= "+(3.14*r*r));

        }
        public void perimeter(){
                Scanner s=new Scanner(System.in);
                System.out.println("Enter the radius");
                int r=s.nextInt();
                System.out.println("Perimeter of Circle= "+(2*3.14*r));
        }
}
class Rectangle implements Inter{
        public void area(){
                Scanner s=new Scanner(System.in);
                System.out.println("Enter the l and b");
                int l=s.nextInt();
                int b=s.nextInt();
                System.out.println("Area of Rectangle= "+(l*b));


        }
        public void perimeter(){
                Scanner s=new Scanner(System.in);
                System.out.println("Enter the l and b");
                int l=s.nextInt();
                int b=s.nextInt();
                System.out.println("Perimeter of Rectangle=
"+(2*(l+b)));
        }
}
```

OUTPUT:

```
1-Area of Circle
2-Perimeter of Circle
3-Area of Rectangle
4-Perimeter of Rectangle
0-Exit
Enter the choice
1
Enter the radius
10
Area of Circle= 314.0
1-Area of Circle
2-Perimeter of Circle
3-Area of Rectangle
4-Perimeter of Rectangle
0-Exit
Enter the choice
2
Enter the radius
10
Perimeter of Circle= 62.800000000000004
1-Area of Circle
2-Perimeter of Circle
3-Area of Rectangle
4-Perimeter of Rectangle
0-Exit
Enter the choice
3
Enter the l and b
3
4
Area of Rectangle= 12
1-Area of Circle
2-Perimeter of Circle
3-Area of Rectangle
4-Perimeter of Rectangle
0-Exit
Enter the choice
4
Enter the l and b
3
4
Perimeter of Rectangle= 14
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-16

AIM: Prepare bill with the given format using calculate method from interface :

Order No.

Date:

Product Id	Name	Quantity	unit price	Total
101	A	2	25	50
102	B	1	100	100

Net. Amount 150

PROGRAM CODE

```
import java.util.Scanner;
import java.util.Date;
public class bill implements outline{
        int id,quantity,unit,total,orderid;
        String  name;
        Date d;
        public void addItem(){
                System.out.println("Enter the item id");
                id=s.nextInt();
                s.nextLine();
                System.out.println("Enter the item name");
                name=s.nextLine();
                System.out.println("Enter the item quantity");
                quantity=s.nextInt();
                System.out.println("Enter the item unit price");
                unit=s.nextInt();
                s.nextLine();
                total=unit*quantity;
        }
        public void forHeader(){
                d=new Date();
                System.out.println("Enter the Order ID");
                orderid=s.nextInt();
                s.nextLine();

        }
        public void showHeader(){
                System.out.println("\nOrder ID : "+orderid);
                System.out.println("\nDate :"+d.toString());
        }
        public void prepareBill(){
                System.out.format("%10d %10s %10d %10d
%10d",id,name,quantity,unit,total);
```

```
        }
        public static void main(String[] args) {
                Scanner s=new Scanner(System.in);
                int ch=1;
                int n=5,i=0,net=0;
                bill newbill[]=new bill[n];
                while(ch==1 && i<n ){
                        newbill[i]=new bill();
                        if(i==0){
                                newbill[i].forHeader();
                        }
                        System.out.println("Ttem "+(i+1));
                        newbill[i].addItem();
                        i++;
                        System.out.println("Enter 1 to add more
items");
                        ch=s.nextInt();         }
                newbill[0].showHeader();
                System.out.printf("%10s %10s %10s %10s
%10s","PRODUCT ID", "NAME", "QUANTITY", "UNIT PRIZE",
"TOTAL");
                System.out.println();
                for(int z=0;z<55;z++){
                        System.out.print("-");          }
                System.out.println();
                for(int j=0;j<i;j++){
                        newbill[j].prepareBill();
                        System.out.println();           }
                for(int z=0;z<55;z++){
                        System.out.print("-");
                }
                System.out.println();
                for(int j=0;j<i;j++){
                        net+=newbill[j].total;          }
                System.out.println("Net Amount :"+net);     }
}
interface outline{
        Scanner s=new Scanner(System.in);
        public void prepareBill();
        void addItem();
        void forHeader();
        void showHeader();}
```

OUTPUT:

```
Enter the Order ID
101
Ttem 1
Enter the item id
1
Enter the item name
pen
Enter the item quantity
10
Enter the item unit price
3
Enter 1 to add more items
1
Ttem 2
Enter the item id
2
Enter the item name
book
Enter the item quantity
5
Enter the item unit price
30
Enter 1 to add more items
0

Order ID : 101

Date :Sun Jul 10 21:57:28 IST 2022
PRODUCT ID       NAME   QUANTITY UNIT PRIZE      TOTAL
---------------------------------------------------------
         1        pen         10          3         30
         2       book          5         30        150
---------------------------------------------------------
Net Amount :180
```

RESULT:

The above program is successfully executed and obtained the output

COURSE OUTCOME-4

PROGRAM-17

AIM: Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

PROGRAM CODE

graphicsimp.java

```
import graphics.*;
public class graphicsimp {
        public static void main(String[] args) {
                circle c1=new circle(3);
                System.out.println("Area of circle with radius 3");
                c1.area();
                rectangle r1=new rectangle(2,3);
                System.out.println("Area of rectangle with length 2 and
breadth 3");
                r1.area();
                triangle t1=new triangle(5,3);
                System.out.println("Area of triangle with b=5 and
heigth 3");
                t1.area();
                square s1=new square(5);
                System.out.println("Area of square with side 5");
                s1.area();
        }
}
```

circle.java

```
package graphics;
public class circle implements inter{
        int r;
        public circle(int i){
                r=i;
        }
        public void area(){
                System.out.println(3.14*r*r);
        }
}
```

rectangle.java

```
package graphics;
public class rectangle implements inter{
	int l,b;
	public rectangle(int i,int j){
		l=i;
		b=j;
	}
	public void area(){
		System.out.println(l*b);
	}
}
```

square.java

```
package graphics;
public class square {
	int a;
	public square(int i){
		a=i;
	}
	public void area(){
		System.out.println(a*a);
	}
}
```

triangle.java

```
package graphics;
public class triangle {
	int b,h;
	public triangle(int i,int j){
		h=j;
		b=i;
	}
	public void area(){
		System.out.println(0.5*b*h);
	}
}
```

inter.java

```
package graphics;
public interface inter {
	void area();
}
```

OUTPUT:

```
Area of circle with radius 3
28.259999999999998
Area of rectangle with length 2 and breadth 3
6
Area of triangle with b=5 and heigth 3
7.5
Area of square with side 5
25
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-18

AIM: Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers

PROGRAM CODE

arithemetic_imp .java

```
import arithemetic.*;
import java.util.Scanner;
public class arithemetic_imp {
	public static void main(String[] args) {
		Scanner s=new Scanner(System.in);
		System.out.println("Enter a and b");
		int a=s.nextInt();
		int b=s.nextInt();
		calculate c1=new calculate(a,b);

		c1.add();
		c1.sub();
		c1.mult();
		c1.div();
	}}
```

arithe_interfac e.java

```
package arithemetic;
public interface arithe_interface {
	void add();
	void sub();
	void mult();
	void div();}
```

calculate.java

```
package arithemetic;
public class calculate implements arithe_interface  {
	int a,b;
	public calculate(int i,int j){
		this.a=i;this.b=j;}
	public void add(){
		System.out.println("Sum="+(a+b));
	}public void sub(){
		System.out.println("Difference="+(a-b));
	}
	public void mult(){
		System.out.println("Multiplication="+(a*b));
	}public void div(){
		System.out.println("Division="+(a/b));
	}
}
```

OUTPUT:

```
Enter a and b
20
5
Sum=25
Difference=15
Multiplication=100
Division=4
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-19

AIM: Write a user defined exception class to authenticate the user name and password.

PROGRAM CODE

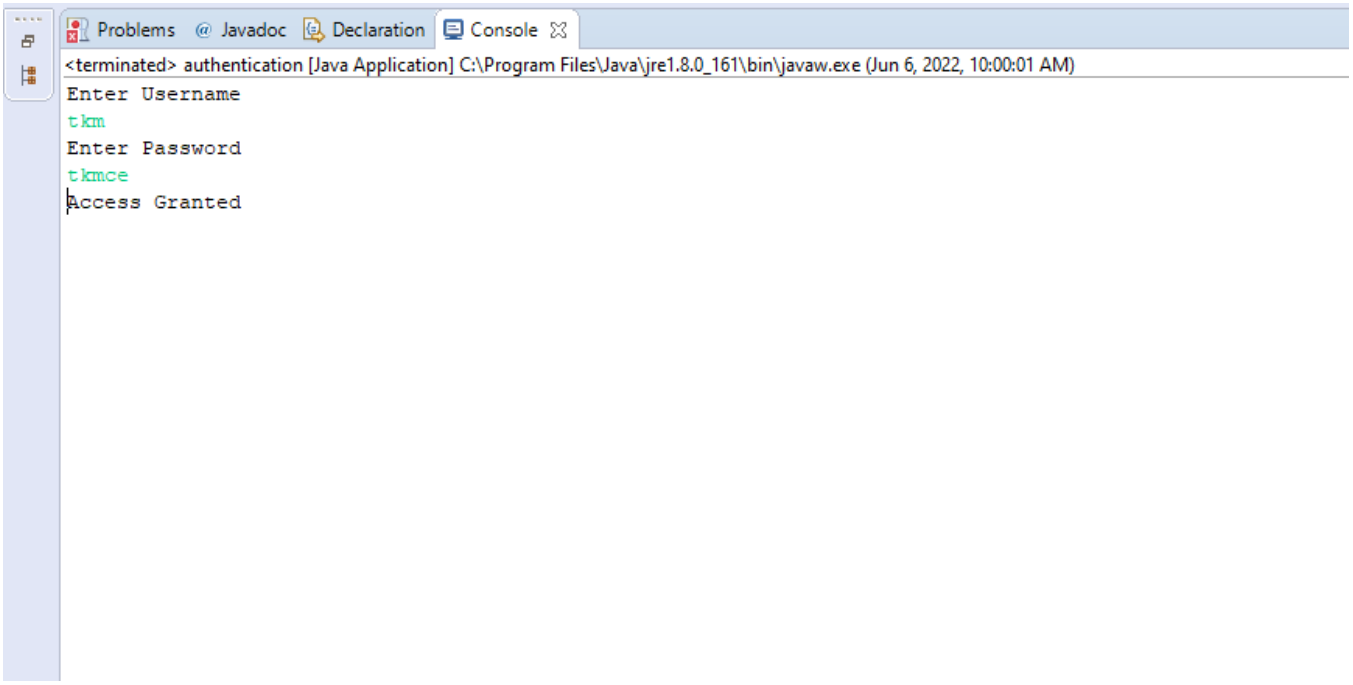
Authentication.java

```
import java.util.*;
public class authentication {
	public static void main(String args[]) {
		String usr = "tkm";
		String pass = "tkmce";
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter Username");
		String username = sc.nextLine();
		System.out.println("Enter Password");
		String password = sc.nextLine();
		try {
			if((username.equals(usr)) &&
(password.equals(pass))){
				System.out.println("Access Granted");
			}
			else {
				throw new credential("Invalid
Credentials");
			}
		}catch(credential e) {
			System.out.println(e.getMessage());
		}

	}
}
```

credential.java

```
public class credential extends Exception {
	public credential(String s) {
		super(s);
	}
}
```

OUTPUT:



RESULT:

The above program is successfully executed and obtained the output

PROGRAM-20

AIM: Find the average of N positive integers, raising a user defined exception foreach negative input.

PROGRAM CODE

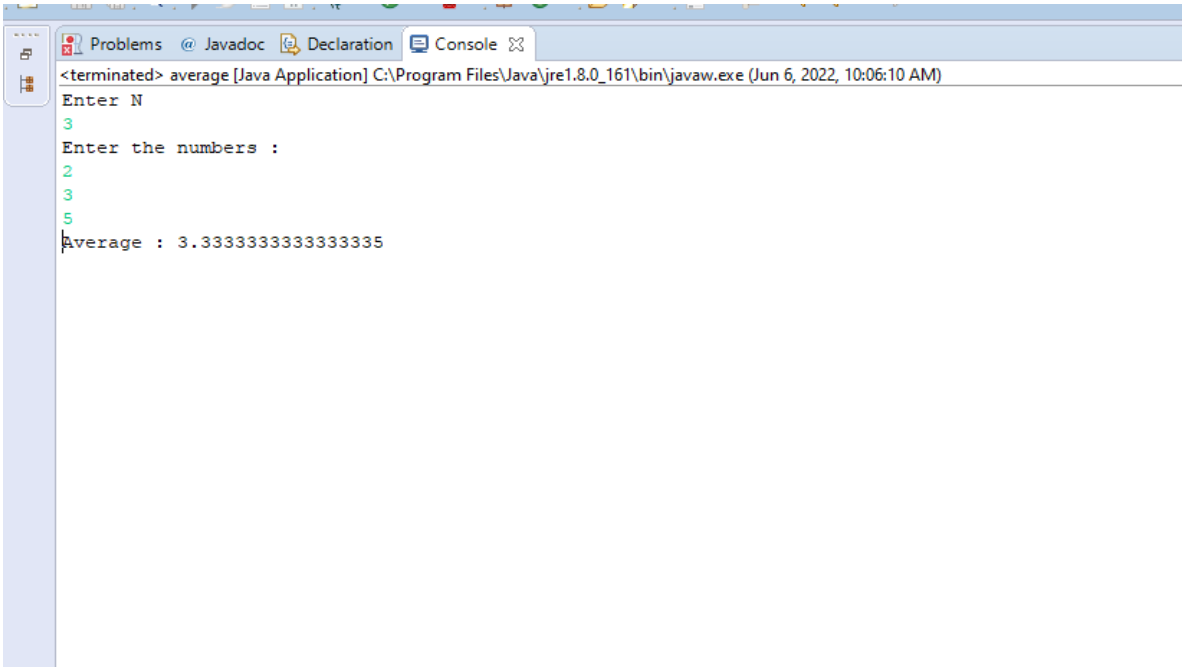
average.java

```
import java.util.*;
public class average {
	public static void main(String args[]) {
		double sum =0;
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter N");
		int N = sc.nextInt();
		int [] num = new int[N];
		System.out.println("Enter the numbers : ");
		for(int i =0;i<N;i++) {
			num[i] = sc.nextInt();
		}
		for(int i=0;i<N;i++) {
			try {
				if(num[i]>=0) {
					sum+= num[i];
				}else {
					throw new
negexception("Negative number");
				}
			}catch(negexception e) {
				System.out.println(e.getMessage());;
			}
		}
		double avg = sum/N;
		System.out.println("Average : "+avg);
	}
}
```

negexception.java

```
public class negexception extends Exception {
	public negexception(String s) {
		super(s);
	}

}
```

OUTPUT:



RESULT:

The above program is successfully executed and obtained the output

PROGRAM-21

AIM: Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

PROGRAM CODE

```
import java.util.Scanner;
class Prime extends Thread
{
        public synchronized void run()
          {
            int i = 0;
            int num = 0;
            String primeNumbers = "";

            for (i = 1; i <= 10; i++) {
              int counter = 0;
              for (num = i; num >=1; num--)
                if (i % num == 0)
                  counter = counter + 1;
                }
              }
              if (counter == 2) {
                primeNumbers = primeNumbers + i + " ";
              }
            }
           System.out.println("\nPrime numbers from 1-10 : \n"
              + primeNumbers);

            System.out.println();
          }}
class FiveTable extends Thread
{
        public synchronized void run()
        {
                System.out.println("Multiplication Table of 5");
                for(int i=1;i<=10;++i)
                {
                        System.out.println("5 * "+i+"="+(5*i));
                }}}
public class ThreadSync {
        public static void main(String args[])
        {
                Scanner sc=new Scanner(System.in);
                Prime p=new Prime();
      p.start();
                FiveTable f=new FiveTable();
                f.start();
        }
}
```

OUTPUT:

```
Prime numbers from 1-10 :
2 3 5 7

Multiplication Table of 5
5 * 1=5
5 * 2=10
5 * 3=15
5 * 4=20
5 * 5=25
5 * 6=30
5 * 7=35
5 * 8=40
5 * 9=45
5 * 10=50
```

RESULT:

The above program is successfully executed and obtained the output.

PROGRAM-22

AIM: Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)

PROGRAM CODE

```
import java.util.Scanner;
public class runint{
	public static void main(String[] args){
		Scanner s=new Scanner(System.in);
System.out.println("Enter the start of even number range");
		int st=s.nextInt();
System.out.println("Enter the limit of even numbers range:");
		int l=s.nextInt();
System.out.println("Enter the limit of fibonnaci series:");
		int f=s.nextInt();
		even obj1=new even(st,l);
		fib obj2=new fib(f);
		Thread t1=new Thread(obj1);
		Thread t2=new Thread(obj2);
		t1.start(); t2.start();}}
class even implements Runnable{
	int m,n;
	public even(int i,int j){
		m=i;
		n=j;		}
	public synchronized void run(){
		System.out.println("Even Numbers from
"+m+"\t"+n);
		for(int i=m;i<=n;i++){
			if(i%2==0){
				System.out.println(i);
			}}}}
class fib implements Runnable{
	int n;
	public fib(int j){
		n=j;}
	public synchronized void run(){
		System.out.println("Fibonacci series upto "+n);
		int a=0,b=1,c=0;
		System.out.println(a+" "+b);
		while((a+b)<n){
			c=a+b;
			System.out.println(c);
			a=b;b=c;
}}}
```

OUTPUT:

```
Enter the start of even number range
2
Enter the limit of even numbers range:
15
Enter the limit of fibonnaci series:
8
Fibonacci series upto 8
0
1
1
2
3
5
Even Numbers from 2     15
2
4
6
8
10
12
14
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-23

AIM: Producer/Consumer using ITC

PROGRAM CODE

```
import java.util.ArrayList;
class Producer1 implements Runnable{
	ArrayList<Integer>l;
	int i=0;
	Producer1(ArrayList<Integer> l){
		this.l=l;
	}
	public void run() {
		try {while(true) {
				produce1(i++);
				if((i)==20) {
					break;}}}
		catch(Exception e) {
			System.out.println(e.getMessage());
		}}
	public void produce1(int i) throws Exception {
		synchronized(l)
		{ System.out.println("produced:"+i);
			l.add(i);
			l.notify();}
		synchronized(l) {
			while(l.size()==5) {
				System.out.println("Production
Full");
				l.wait();
			}}}}
class Consumer1 implements Runnable{
	ArrayList<Integer>l;
	Consumer1(ArrayList<Integer>l)
	{
	this.l=l;
	}
```

```
        public void run() {
                while(true)
                { try
                        {consume1();
                        }
                        catch (Exception e)
                        {
                                System.out.println(e.getMessage());
                        }}}
        public void consume1() throws Exception
        {synchronized(l)
                {while(l.isEmpty())
                        {System.out.println("fully Consumed");
                                l.notify();
                                Thread.sleep(500);
                                l.wait();
                        }}
                synchronized(l)
                {
                        Thread.sleep(500);
                System.out.println("Consumed"+l.remove(0));
                        }}}
public class co4_pg7{
        public static void main(String args[]) {
                ArrayList<Integer>l=new ArrayList<>();
                Producer1 obj=new Producer1(l);
                Thread t1=new Thread(obj);
                Consumer1 obj2=new Consumer1(l);
                Thread t2=new Thread(obj2);
                t1.start();
                t2.start();
        }
}
```

OUTPUT:

```
Production Full
Consumed0
Consumed1
Consumed2
Consumed3
Consumed4
fully Consumed
produced:5
produced:6
produced:7
produced:8
produced:9
Production Full
Consumed5
Consumed6
Consumed7
Consumed8
Consumed9
fully Consumed
produced:10
produced:11
produced:12
produced:13
produced:14
Production Full
Consumed10
Consumed11
Consumed12
Consumed13
Consumed14
fully Consumed
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-24

AIM: Program to create a generic stack and do the Push and Pop operations.

PROGRAM CODE

```
import java.util.*;
class Stack <T>{
	ArrayList<T> S;
	int top=-1,size;
	Stack(int s){
		this.size=s;
		this.S=new ArrayList<T>(size);
	}
	void push(T newData) {
		if(top+1 == size) {
			System.out.println("Stack overflow");
		}
		else {
			top++;
			if(S.size()>top) {
				S.set(top,newData);
			}
			else {
				S.add(newData);
			}}}
		void pop() {
		if(top==-1) {
			System.out.println("Stack Underflow");
		}
		else {
			top--;
		}}
	void display() {
		for(int i=0;i<=top;i++) {
			System.out.println(S.get(i));
		}}
```

```
        T top() {
                if(top==-1) {
                        System.out.println("Stack Underflow");
                        return null;
                }
                else {
                        return S.get(top);
                }}}
public class genimp{
        public static void main(String args[]) {
                Stack<Integer> obj=new Stack<>(5);
                obj.push(10);
                obj.push(20);
                obj.push(30);
                obj.push(40);
                obj.push(50);
                System.out.println("After Push");
                obj.display();
                obj.pop();
                obj.pop();
                obj.pop();
                System.out.println("After Pop");
                obj.display();
                System.out.println("Top");
                System.out.println(obj.top());
        }
}
```

OUTPUT:

```
After Push
10
20
30
40
50
After Pop
10
20
Top
20
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-25

AIM: Using generic method perform Bubble sort.

PROGRAM CODE

```
import java.util.Scanner;
public class bs{
public static void main(String[] args){
                Scanner s=new Scanner(System.in);
System.out.println("Enter the number of elements in the array");
                int l=s.nextInt();
                Integer[] arr=new Integer[l];
                System.out.println("Enter the elements");
                for(int i=0;i<l;i++){arr[i]=s.nextInt();
                }Bubblesort<Integer> b1=new Bubblesort<>(arr);
                System.out.println("Orginal Array");
                b1.display();b1.sort();
                System.out.println("Array after bubblesort");
                b1.display();}}
class Bubblesort<T extends Comparable<? super T>>{
        T array[];int n;
        Bubblesort(T a[]){this.array=a;}
        void display(){System.out.println();
                for(int i=0;i<array.length;i++){
                        System.out.print(array[i]+", ");}}
        void sort(){
                for(int i=0;i<array.length-1;i++){
                        for(int j=0;j<array.length-n-1;j++){
                        if(array[j].compareTo(array[j+1])>0){
                                        swap(j,array); }}}}
        void swap(int j,T[] array){
                T temp=array[j];
                array[j]=array[j+1];
                array[j+1]=temp;
        }}
```

OUTPUT:

```
Enter the number of elements in the array
4
Enter the elements
1
8
6
2
Orginal Array

1, 8, 6, 2, Array after bubblesort

1, 2, 6, 8,
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-26

AIM: Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

PROGRAM CODE

```
import java.util.*;
public class ArrList{
        public static void main(String[] args){
                ArrayList<String> cars=new ArrayList<>();
                Scanner s=new Scanner(System.in);
                int ch,index;tring name;
System.out.println("1-add\n2-remove\n3-display\n4-Exit\n");
                do{System.out.println("Enter the choice");
                        ch=s.nextInt(); s.nextLine();
                        switch(ch){
                                case 1: System.out.println("Enter
the car name");name=s.nextLine();
                cars.add(name);
                break;
case 2: System.out.println("Enter the index to remove item");
index=s.nextInt(); s.nextLine();
cars.remove(index);
break;
case 3:
System.out.println("\nCars are :");
        for(String i:cars){ System.out.println(i);}
                                        break;}
                }while(ch!=4);}
}
```

OUTPUT:

```
1-add
2-remove
3-display
4-Exit

Enter the choice
1
Enter the car name
abc
Enter the choice
1
Enter the car name
xxx
Enter the choice
3

Cars are :
abc
xxx
Enter the choice
2
Enter the index to remove item
1
Enter the choice
3

Cars are :
abc
Enter the choice
4
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-27

AIM: Program to remove all the elements from a linked list

PROGRAM CODE

```
import java.util.*;
public class linkedimp{
        public static void main(String[] args){
                Scanner s=new Scanner(System.in);
                linked<String> l1=new linked<>();
                int ch;String str;
                do{System.out.println("\n1-insert\n2-display\n3-clear LL\n4-Exit\nEnter the choice");
                        ch=s.nextInt(); s.nextLine();
                        switch(ch){
                                case 1:System.out.println("Enter the String");
                                        str=s.nextLine();
                                        l1.insert(str);
                                        break;
                                case 2:l1.display();break;
                                case3:l1.clear();l1.display();break;
                        }}while(ch!=4);
        }}class linked<T>{
LinkedList<T> ll=new LinkedList<>();
        void insert(T data){
                ll.add(data);}
        void display(){System.out.println(ll);
        }void clear(){
ll.clear();}
        }
```

OUTPUT:

```
1-insert
2-display
3-clear LL
4-Exit
Enter the choice
1
Enter the String
hello

1-insert
2-display
3-clear LL
4-Exit
Enter the choice
2
[hello]

1-insert
2-display
3-clear LL
4-Exit
Enter the choice
3
[]

1-insert
2-display
3-clear LL
4-Exit
Enter the choice
4
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-28

AIM: Program to remove an object from the Stack when the position is passed as parameter

PROGRAM CODE

```
import java.util.*;import java.io.*;
public class stackpos{
        public static void main(String[] args){
                Stack<String> st=new Stack<>();
                Scanner s=new Scanner(System.in);
                String str;
System.out.println("Enter the number of items");
                int n=s.nextInt(); s.nextLine();
                System.out.println("Enter the elements");
                for(int i=0;i<n;i++){
                        str=s.nextLine();
                        st.push(str);
                }
    System.out.println(st);
System.out.println("Enter the index of the element to be deleted");
int in=s.nextInt();
                st.remove(in);
                System.out.println(st);
        }
}
```

OUTPUT:

```
Enter the number of items
4
Enter the elements
1
5
9
8
[1, 5, 9, 8]
Enter the index of the element to be deleted
2
[1, 5, 8]
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-29

AIM: Program to demonstrate the creation of queue object using the PriorityQueue class

PROGRAM CODE

```
import java.util.*;
public class pq {
public static void main(String[] args) {
                int n;String str;
        PriorityQueue<String> pqueue=new PriorityQueue<>();
                System.out.println("Total count");
                Scanner s=new Scanner(System.in);
                n=s.nextInt();s.nextLine();
                System.out.println("Enter data");
                for(int i=0;i<n;i++){
                        str=s.nextLine();
                        pqueue.add(str);}
                System.out.println("Peek: "+pqueue.peek());
                System.out.println("Data in Queue");
                Iterator<String> itr1=pqueue.iterator();
                while(itr1.hasNext()){
                System.out.println(itr1.next());  }
System.out.println("Polling: "+pqueue.poll());
System.out.println("After polling data in Queue");
Iterator<String> itr2=pqueue.iterator();
                while(itr2.hasNext()){
                System.out.println(itr2.next());
                }
        }}
```

OUTPUT:

```
Total count
5
Enter data
78
42
12
54
69
Peek: 12
Queue
12
54
42
78
69
Polling: 12
After polling data in Queue
42
54
69
78
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-30

AIM: Program to demonstrate the addition and deletion of elements in deque

PROGRAM CODE	import java.util.*;
	public class co4_pg14 {
	public static void main(String[] args){
	Deque<Integer> dq=new LinkedList<>();
	dq.add(1);
	dq.add(2);
	dq.addFirst(3);
	dq.addLast(4);
	dq.push(5);
	dq.offer(6);
	dq.offerFirst(7);
	System.out.print("DEQUE: "+dq+" ");
	dq.removeFirst();
	System.out.println("\nDEQUE after removing first element");
	System.out.print(dq+" ");
	dq.removeLast();
	System.out.println("\nDEQUE after removing last element");
	System.out.print(dq+" ");}}

OUTPUT:

```
DEQUE:  [7, 5, 3, 1, 2, 4, 6]
DEQUE after removing first element
[5, 3, 1, 2, 4, 6]
DEQUE after removing last element
[5, 3, 1, 2, 4]
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-31

AIM: Program to demonstrate the creation of Set object using the LinkedHashsetclass

PROGRAM CODE

```
import java.util.*;

public class lhash {

        public static void main(String args[]) {

                LinkedHashSet<String> ln = new
LinkedHashSet<>();

                System.out.println("Adding Elements to the
linkedHashSet : ");

                ln.add("Java");

                ln.add("Python");

                ln.add("MongoDB");

                for(String i :ln ) {System.out.print(i+" ");}

                System.out.println("\nRemoving the element
Python from the linkedHashSet : ");

                ln.remove("Python");System.out.println(ln);

                System.out.println("\nTrying to remove C which
is not present : "+ln.remove("C"));

System.out.println("\nChecking if Java  contains :  "+
ln.contains("Java"));

        }}
```

OUTPUT:

```
Adding Elements to the linkedHashSet :
Java Python MongoDB
Removing Elements to the linkedHashSet :
[Java, MongoDB]

Trying to remove C which is not present : false

Checking if Java  contains :  true
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-32

AIM: Write a Java program to compare two hash set.

PROGRAMCODE

```
import java.util.*;
public class hset{
        public static void main(String[] args){
HashSet<String> set1=new HashSet<>();
HashSet<String> set2=new HashSet<>();
HashSet<String> set3=new HashSet<>();
                set1.add("A");set1.add("B");
                set1.add("C");set1.add("D");
                set2.add("A");set2.add("B");
                set2.add("C");set2.add("D");
               set3.add("A");set3.add("B");
                set3.add("C");set3.add("D");
                set3.add("E");
System.out.println("1: "+set1+"\n2:
"+set2+"\n3: "+set3);
                System.out.println("Both sets
1 and 2 are equal: "+set1.equals(set2));
                System.out.println("Both sets
1 and 3 are equal: "+set1.equals(set3));
        }
}
```

OUTPUT:

```
1: [A, B, C, D]
2: [A, B, C, D]
3: [A, B, C, D, E]
Both sets 1 and 2 are equal: true
Both sets 1 and 3 are equal: false
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-33

AIM: Program to demonstrate the working of Map interface by adding, changingand removing elements.

PROGRAM CODE

```
import java.util.*;
public class map{
        public static void main(String[] args){
        Map<Integer,String> mapp=new
HashMap<Integer,String>();
        mapp.put(3,"Java");
        mapp.put(2,"C");
        mapp.put(1,"Python");
        System.out.println("Map: "+mapp);
        mapp.put(2,"C++");
        mapp.put(null,"SQL");
        System.out.println("Map after updation: "+mapp);
        mapp.remove(3);
        System.out.println("Map after deletion: "+mapp);
        System.out.println("\nAccessing using Map.Entry
Interface");
        for (Map.Entry<Integer, String> me :mapp.entrySet()){
                System.out.println(me.getKey() + ":"
+me.getValue() );
        }
        }
}
```

OUTPUT:

```
Map: {1=Python, 2=C, 3=Java}
Map after updation: {null=SQL, 1=Python, 2=C++, 3=Java}
Map after deletion: {null=SQL, 1=Python, 2=C++}

Accessing using Map.Entry Interface
null:SQL
1:Python
2:C++
```

RESULT:

The above program is successfully executed and obtained the output

PROGRAM-34

AIM: Program to Convert HashMap to TreeMap

PROGRAM CODE

```
import java.util.*;
public class hash2tree{
        public static void main(String[] args){
                Map<Integer,String> hm=new HashMap<>();
                hm.put(1,"One");
                hm.put(2, "Two");
                hm.put(15, "Fifteen");
                hm.put(7, "Seven");
                hm.put(3, "Three");
                hm.put(9, "Nine");
                System.out.println("Hashmap: "+hm);
                Map<Integer,String> tm=new TreeMap<>();
                tm.putAll(hm);
                System.out.println("\nTreemap from Hashmap:
"+tm);

        }
}
```

OUTPUT:

```
Hashmap: {1=One, 2=Two, 3=Three, 7=Seven, 9=Nine, 15=Fifteen}

Treemap from Hashmap: {1=One, 2=Two, 3=Three, 7=Seven, 9=Nine, 15=Fifteen}
```

RESULT:

The above program is successfully executed and obtained the output

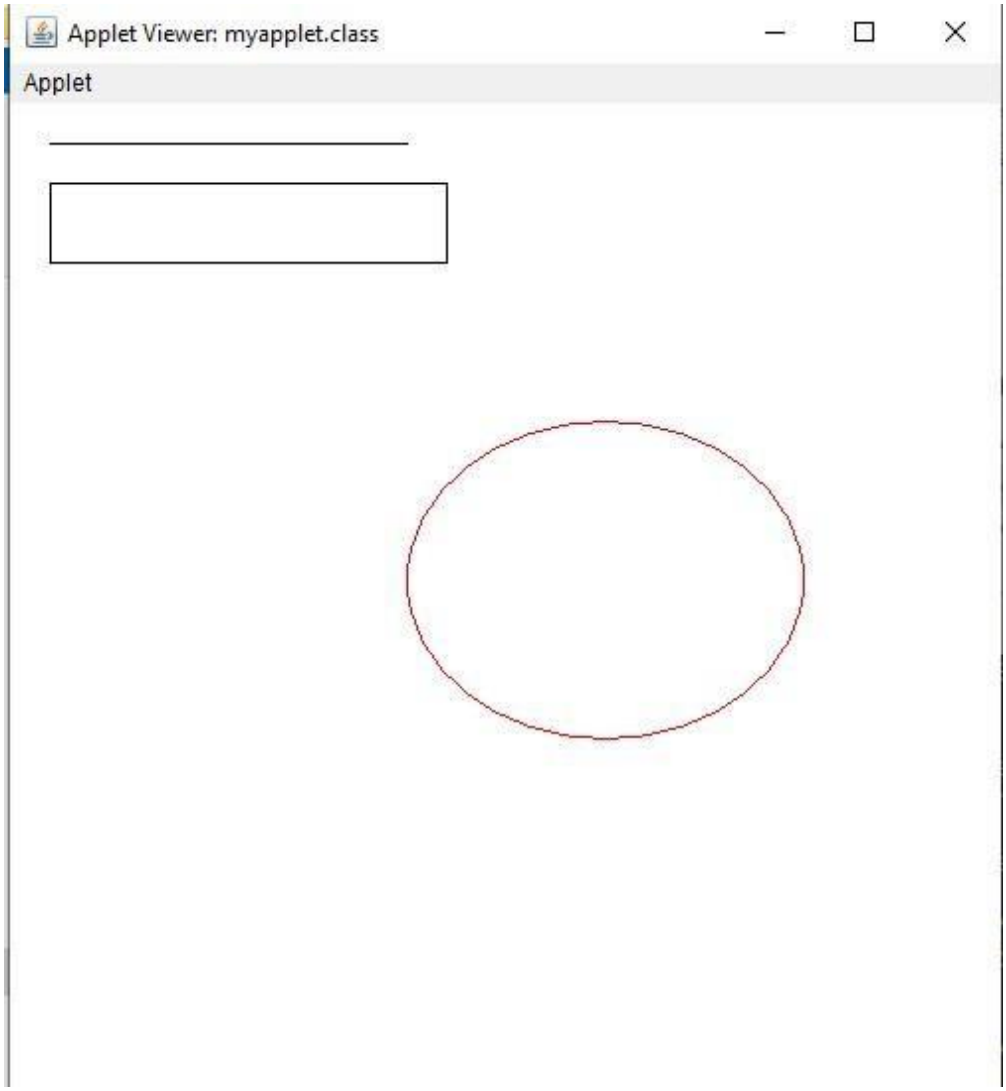
COURSE OUTCOME-5

PROGRAM-35

AIM: Program to draw Circle, Rectangle, Line in Applet.

PROGRAM CODE	import java.applet.*;
	import java.awt.*;
	import java.awt.Graphics;
	public class myapplet extends Applet
	{
	public void paint(Graphics g){
	g.drawLine(20, 20, 200, 20);
	g.drawRect(20, 40, 200, 40);
	g.setColor(Color.RED);
	g.drawOval(200, 160, 200, 160);
	
	}}
	/*
	<applet code="myapplet.class" width="500" height="700" border="2">
	</applet>
	*/

OUTPUT:



RESULT:

The above program is successfully executed and obtained the output

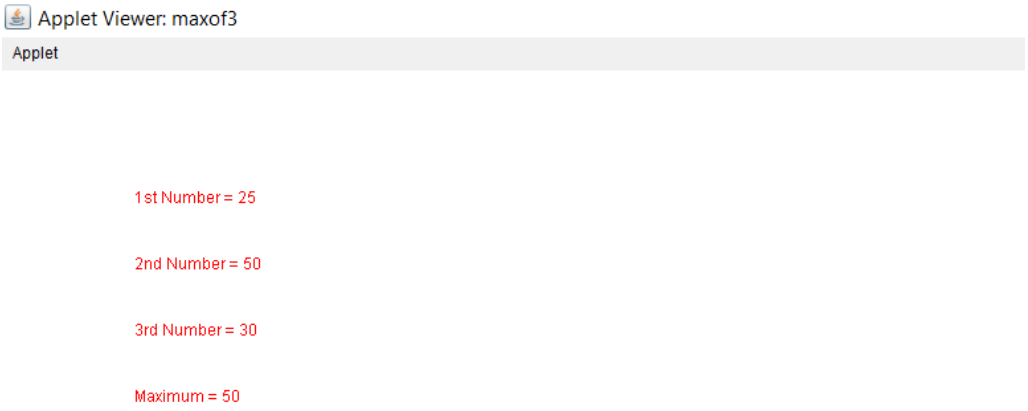
PROGRAM-36

AIM: Program to find maximum of three numbers using AWT

PROGRAM CODE

```
import java.awt.*;
import java.applet.*;
import java.io.*;
/*<applet code="maxof3" width=500 height=500>
<param name="a" value="25">
<param name="b" value="50">
<param name="c" value="30"></applet>*/
public class maxof3 extends Applet{
int a; int b; int c;int d;String str;
public void start()
{String s1;
s1 = getParameter("a"); a = Integer.parseInt(s1);
s1 = getParameter("b"); b = Integer.parseInt(s1);
s1 = getParameter("c");c = Integer.parseInt(s1);}
public void paint(Graphics g){ if( a >= b && a >= c) d = a;
else if (b >= a && b >= c) d=b;
else
d=c;
g.setColor(Color.red);
g.drawString("1st Number = " + a, 100,100);
g.drawString("2nd Number = " + b, 100,150);
g.drawString("3rd Number = " + c, 100, 200);
g.drawString("Maximum = "  + d,  100,250);}}
```

OUTPUT:



RESULT:

The above program is successfully executed and obtained the output

PROGRAM-37

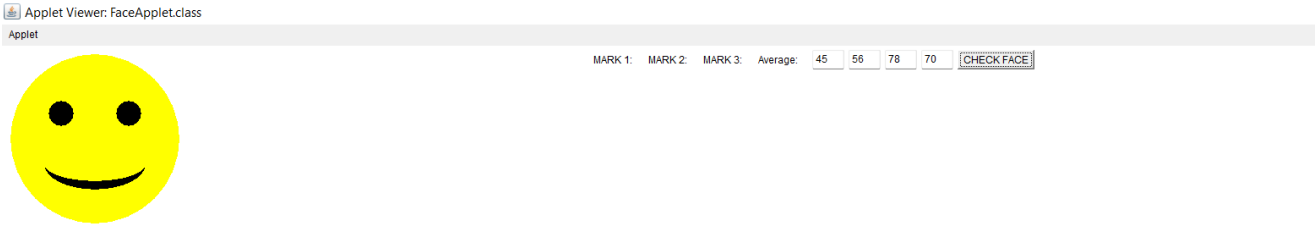
AIM: Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise

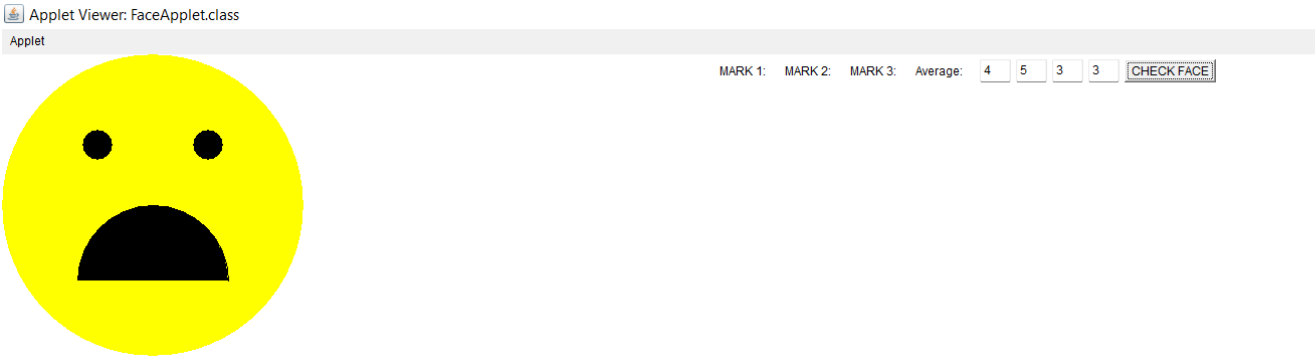
PROGRAM CODE

```
Import java.applet.*;
Import java.awt.*;
public class FaceApplet extends
Applet implements ActionListener
{
Label l1=new Label("MARK 1:");
Label l2=new Label("MARK 2:");
 Label l3=new Label("MARK 3:");
  Label l4=new Label("Average:");
TextField t1=new TextField();
 TextField t2=new TextField();
 TextField t3=new TextField();
 TextField t4=new TextField();
Buttonb=new
Button("CHECKFACE");
public void init()
{
add(l1);add(l2);
add(l3);add(l4);
add(t1);add(t2);
add(t3);add(t4);
add(b);
b.addActionListener(this);
}
Public                    void
actionPerformed(ActionEvent e)
{
int n1=Integer.parseInt(t2.getText());
int n2=Integer.parseInt(t3.getText());
int n3=Integer.parseInt(t3.getText());
if(e.getSource()==b)
{
int avg=(n1+n2+n3)/3;
t4.setText(String.valueOf(avg));
}
}
public void paint(Graphics g)
{
int n1= Integer.parseInt(t1.getText());
int n2= Integer.parseInt(t2.getText());
int n3= Integer.parseInt(t3.getText());
int avg=(n1+n2+n3)/3;
```

```
if(avg > 50)
{

g.setColor(Color.YELLOW);
g.fillOval(10, 10, 200, 200);
g.setColor(Color.BLACK);
g.fillOval(55, 65, 30, 30);
g.fillOval(135, 65, 30, 30);
g.fillOval(50, 110, 120, 60);
g.setColor(Color.YELLOW);
g.fillRect(50, 110, 120, 30);
g.fillOval(50, 120, 120, 40);
}
else
{
g.setColor(Color.yellow);
g.fillOval(0,0,300,300);
g.setColor(Color.black                    );
g.fillOval(80,75,30,30);//sad face
g.fillOval(190,75,30,30);
g.setColor(Color.black);
 g.drawArc(75,150,150,150,0,180);
  g.fillArc(75,150,150,150,0,180);

}
}
}
/*
<applet
          code="FaceApplet.class"
          width="400"       height="400"
border="2">
</applet>
*/
```

OUTPUT:





RESULT:

The above program is successfully executed and obtained the output

PROGRAM-38

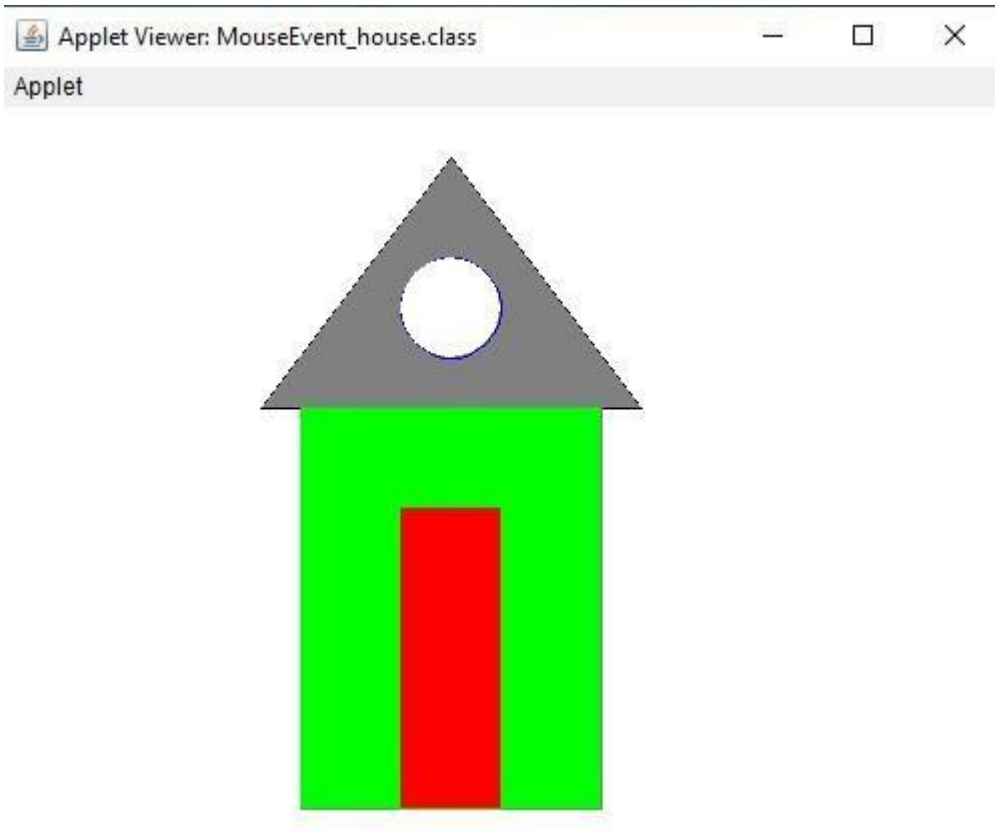
AIM: Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

PROGRAM CODE

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


public class MouseEvent_house extends Applet implements MouseListener
{
   int a,b;
   public void init()
   {
addMouseListener( this);
   }
   public void paint(Graphics g)
   {
   int x[]={130,320,225};
   int y[]={150,150,25};
   g.drawPolygon(x,y,3);
   g.setColor(Color.gray);
   g.fillPolygon(x,y,3);
   g.drawRect(150,150,15
   0,200);//House
   g.setColor(Color.gree);
g.fillRect(150,150,150,200);
g.drawRect(200, 200,50,150);//Door
g.setColor(Color.blue);
 g.fillRect(200,200,50,150);
   g.drawOval(200,75,50,50);
   g.setColor(Color.white);
   g.fillOval(200,75,50,50);
if(a>200 && a<300 && b>200 && b<300)
       {
g.setColor(Color.red);
```

```
g.fillRect(200, 200, 50, 150);
        }
    }
    public void mouseClicked(MouseEvent e)
    {}
    public void mouseEntered(MouseEvent e)
    {}
@Override
    public void mouseExited(MouseEvent e) {}
    public void mousePressed(MouseEvent e)
    {
       a=e.getX();
       b=e.getY();
       repaint();

    }
    public void mouseReleased(MouseEvent e)
    {


    }


 }
 /*
 <applet code="MouseEvent_house.class" width="500"
 height="700" border="2">
 </applet>
       */
```

OUTPUT:



RESULT:

The above program is successfully executed and obtained the output

PROGRAM-39

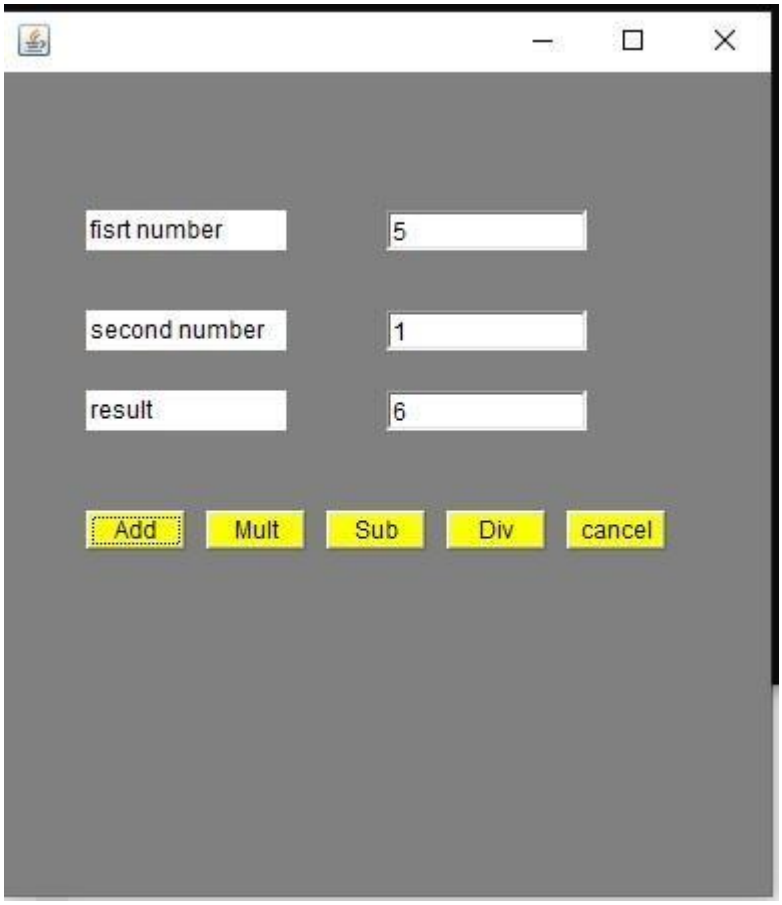
AIM: Implement a simple calculator using AWT components

PROGRAM CODE

```
import java.awt.*;

import java.awt.event.*;

class calculator implements
ActionListener{f=new Frame();

Label l1=new Label("fisrt number");
Label l2=new Label("second
number"); Label l3=new
Label("result");

TextField t1=new TextField();

TextField t2=new TextField();

TextField t3=new TextField();

Button b1=new Button("Add");
Button b2=new Button("Mult");
Button b3=new Button("Sub");
Button b4=new Button("Div");
Button b5=new Button("cancel");

calculator()

{f.add(l1);f.add(l2);

f.add(l3);f.add(t1);

f.add(t2);f.add(t3);

f.add(b1);f.add(b2);f.add(b3);

f.add(b4);f.add(b5);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
  f.setLayout(null); f.setVisible(true);
  f.setSize(400,450);
  f.setLocation(500,200);
  f.setBackground(Color.gray);
  }
  public void actionPerformed(ActionEvent
  e)
  {
```

```
int n1=Integer.parseInt(t1.getText());
int n2=Integer.parseInt(t2.getText());
   if(e.getSource()==b1)
{
t3.setText(String.valueOf(n1+n2));
}
if(e.getSource()==b3)
{
t3.setText(String.valueOf(n1-n2));
}
if(e.getSource()==b2)
{
t3.setText(String.valueOf(n1*n2));
}
if(e.getSource()==b4)
{
t3.setText(String.valueOf(n1/n2));
}if(e.getSource()==b5)
{
System.exit(0);
}}public static void main(String args[])
{
new calculator();
}
}
```

OUTPUT:



RESULT:

The above program is successfully executed and obtained the output

PROGRAM-40

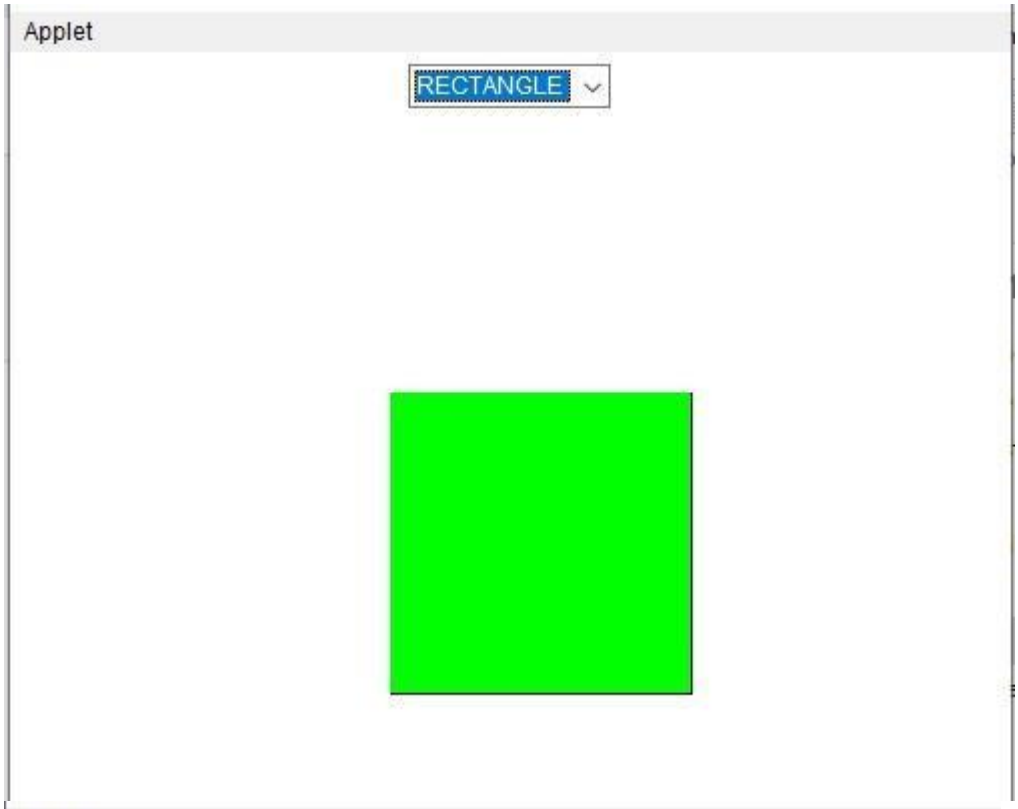
AIM: Develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user's choice.

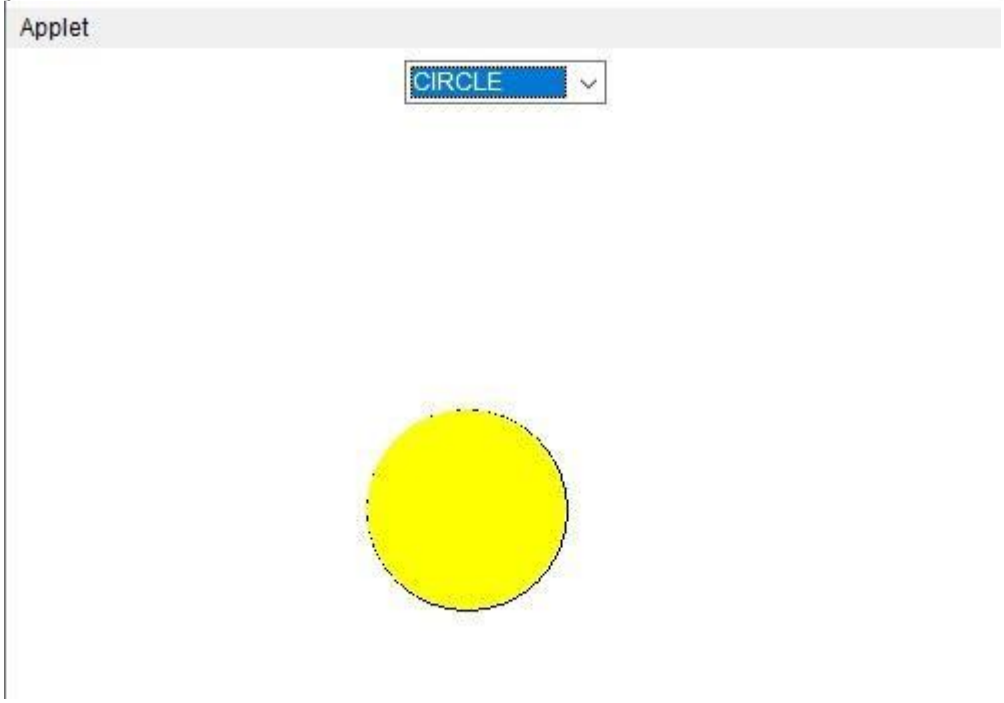
PROGRAM CODE

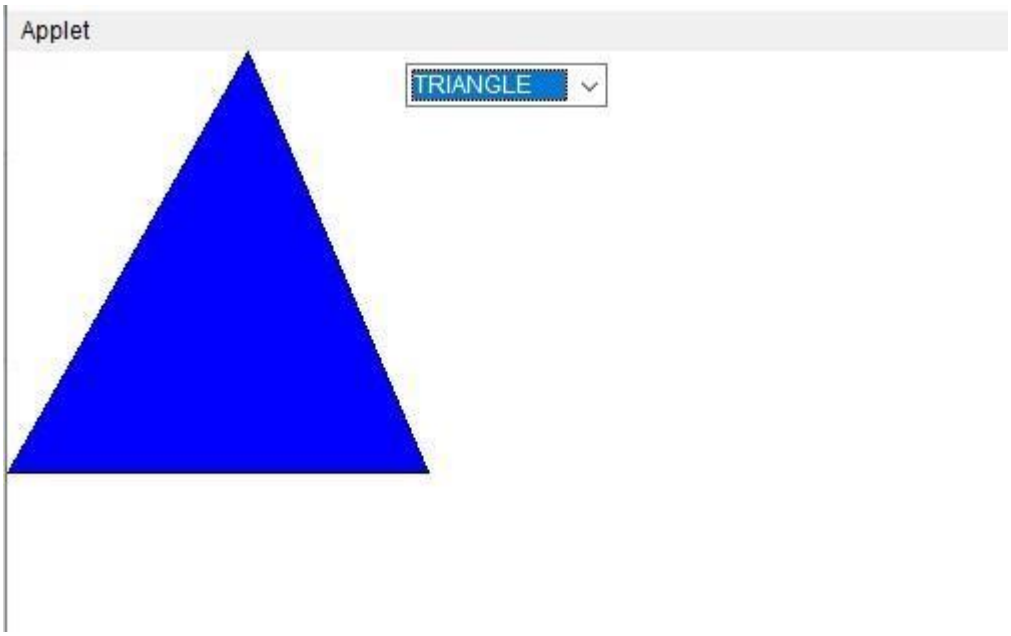
```
import java.applet.*;
import java.awt.*;
import java.awt.Graphics;
import java.awt.event.*;
public class Choice_pgm extends Applet implements
ItemListener {
Choice choice;int
c;
  public void init()
    {
      choice = new Choice();
      choice.addItem("Shapes");
      choice.addItem("RECTANGLE");
      choice.addItem("SQUARE");
      choice.addItem("CIRCLE");
      choice.addItem("TRIANGLE");
      add(choice);
      choice.addItemListener(this);
    }
public void itemStateChanged (ItemEvent e)
    {
       c= choice.getSelectedIndex();
       repaint();}
public void paint(Graphics g){
       super.paint(g);
if (c == 1)
       {g.drawRect(190,170,150,150);
```

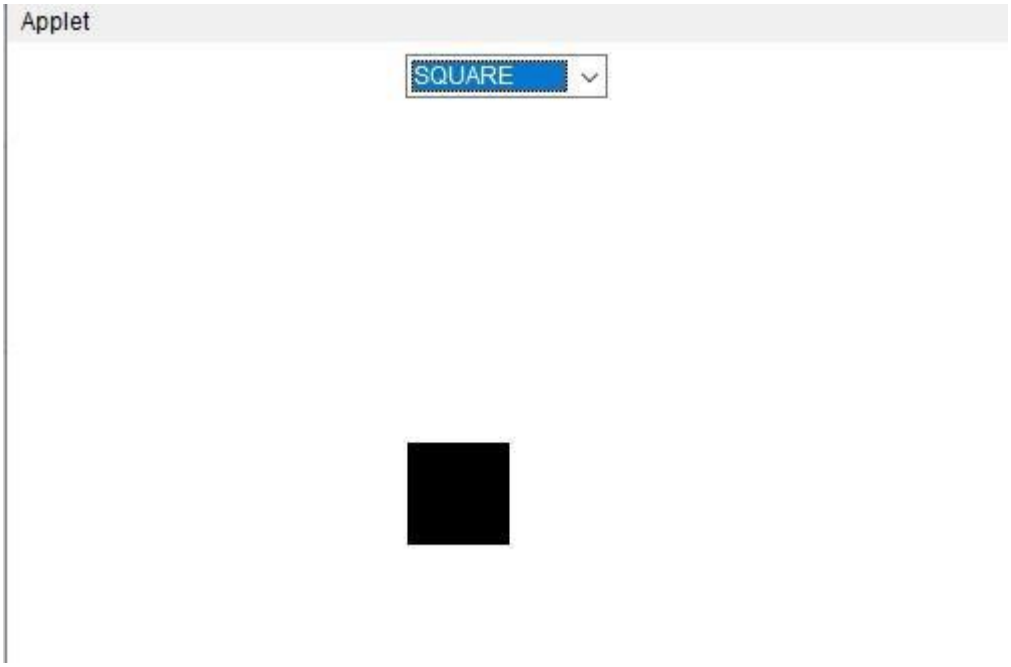
```
            g.setColor(Color.green);
            g.fillRect(190,170,150,150);
          }if (c == 2)
          {g.drawRect(200,200,50,50);g.fillRect(200,200,50,50);
          }
          if (c == 3)
          {g.drawOval(180,180,100,100);
             g.setColor(Color.yellow);
             g.fillOval(180,180,100,100);
} if (c ==4){
             int[] x={120,210,0};
             int[] y={0,210,210};
             g.drawPolygon(x,y,3);
             g.setColor(Color.blue);
             g.fillPolygon(x,y,3);
             }
            }
 }
 /*
 <applet code="Choice_pgm.class" width="500"
height="700" border="2">
 </applet>
         */
```

OUTPUT:









RESULT:

The above program is successfully executed and obtained the output

PROGRAM-41

AIM: Develop a program to handle all mouse events and window events

PROGRAM CODE

```
import java.awt.*;
import java.applet.*;
public class events extends Applet implements MouseListener,MouseMotionListener
{int mx=0;int my=0;
String msg="";
public void init()
{addMouseListener(this); addMouseMotionListener(this);
}
public void mouseClicked(MouseEvent me)
{ mx=20;
my=40;
msg="Mouse Clicked";
repaint();
}
public void mousePressed(MouseEvent me)
{ mx=30;
my=60;
msg="Mouse Pressed";
repaint();}
public void mouseReleased(MouseEvent me)
{mx=30;my=60;
msg="Mouse Released";
repaint();}
public void mouseEntered(MouseEvent me)
{mx=40;my=80;
msg="Mouse Entered";
repaint();}
public void mouseExited(MouseEvent me)
{mx=40;my=80;
msg="Mouse Exited";
repaint();}
```

```
public void mouseDragged(MouseEvent me){
mx=me.getX();
my=me.getY();
showStatus("Currently mouse dragged"+mx+" "+my);
repaint(); }
public void mouseMoved(MouseEvent me)
{ mx=me.getX();
my=me.getY();
showStatus("Currently mouse is at"+mx+" "+my);
repaint();}
public void paint(Graphics g){
g.drawString("Handling Mouse Events",30,20);
g.drawString(msg,60,40);
g.setColor(Color.red);}
}
/*<applet code="events" width=300 height=300>
</applet>*/
```

OUTPUT:

RESULT:

The above program is successfully executed and obtained the output

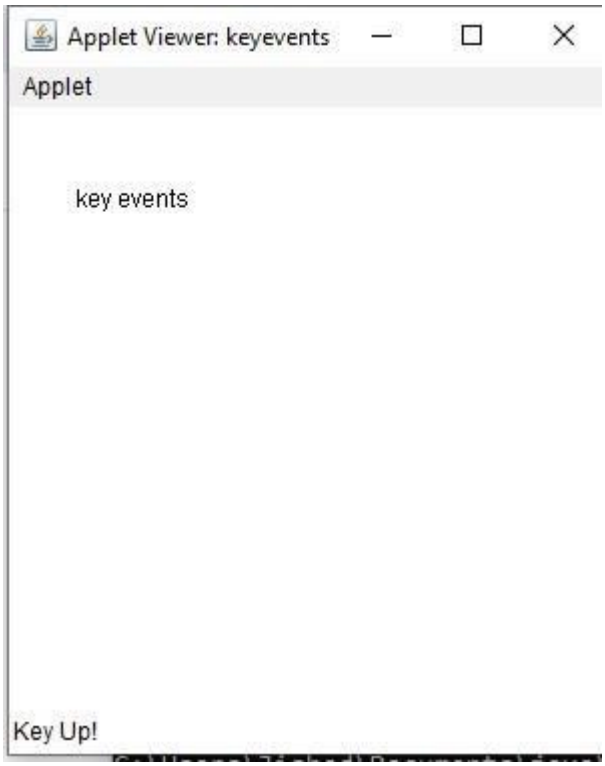
PROGRAM-42

AIM: Develop a program to handle Key events.

PROGRAM CODE

```
import java.awt.*;
import java.applet.*;
/*<applet code="keyevents" width=300 height=300></applet>*/
public class keyevents extends Applet implements KeyListener
{String msg=" "; int x=30,y=50;
 public void init()
{addKeyListener(this);requestFocus();
}
public void keyTyped(KeyEvent ke)
{msg+=ke.getKeyChar();
repaint();
}
public void keyReleased(KeyEvent ke)
{
showStatus("Key Up!");
}
public void keyPressed(KeyEvent ke)
{
showStatus("Key Down!");
}
public void paint(Graphics G)
{
G.drawString(msg,x,y);
}
}
```

OUTPUT:



RESULT:

The above program is successfully executed and obtained the output

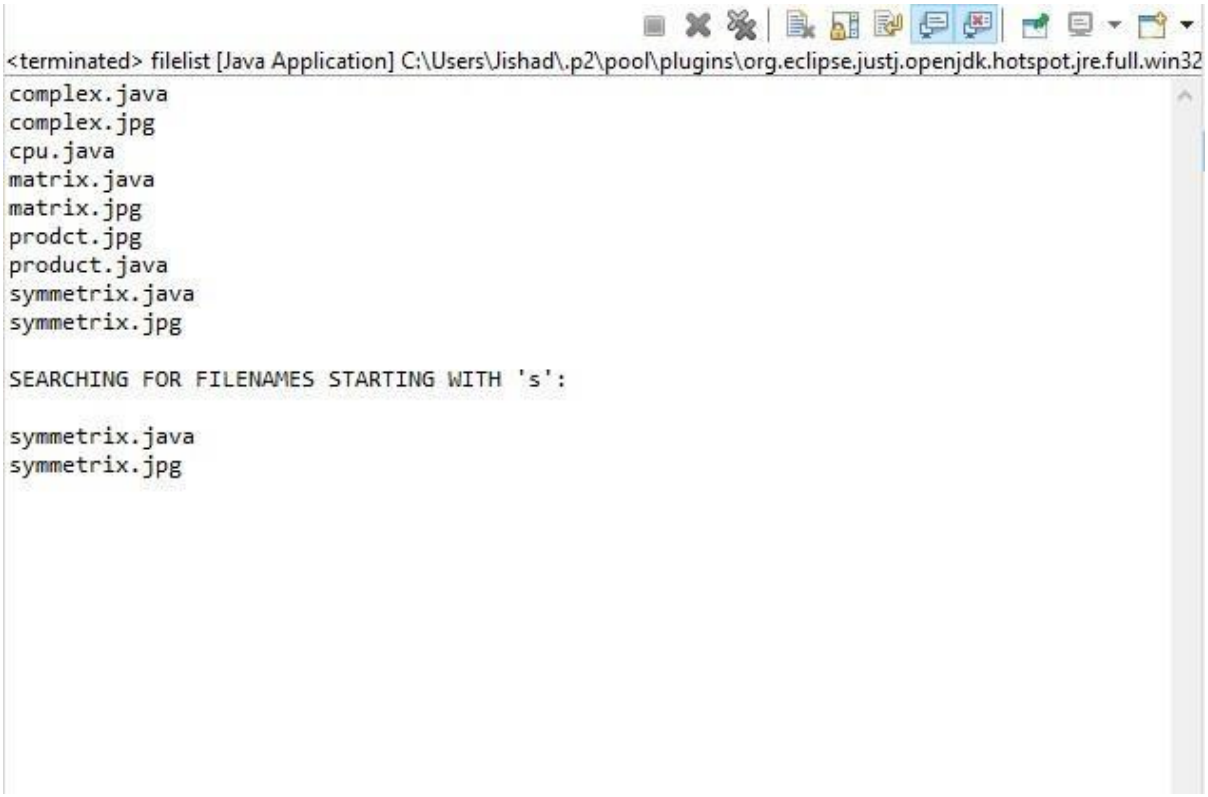
COURSE OUTCOME-6

PROGRAM-43

AIM: Program to list the sub directories and files in a given directory and also search for a file name

PROGRAM CODE	import java.io.*;
	import java.util.*;
	public class filelist{
	public static void main(String[] args) {
	File file = new File("C:\\Users\\Jishad\\Documents\\java\\cycle1 op");
	String[] list = file.list();
	for(String str : list)
	{ System.out.println(str);
	}
	System.out.println("\nSEARCHING FOR FILENAMES STARTING WITH 's':\n");
	FilenameFilter filter = new FilenameFilter() {
	public boolean accept(File dir, String fname) {
	return fname.startsWith("s");
	}
	} String[] search =
	file.list(filter);if(search == null)
	{
	System.out.println("File does not exist.");
	}
	else {
	for(int i=0; i<search.length;i++) {
	String fn = search[i];
	System.out.println(fn);
	}}}}

OUTPUT:



RESULT:

The above program is successfully executed and obtained the output

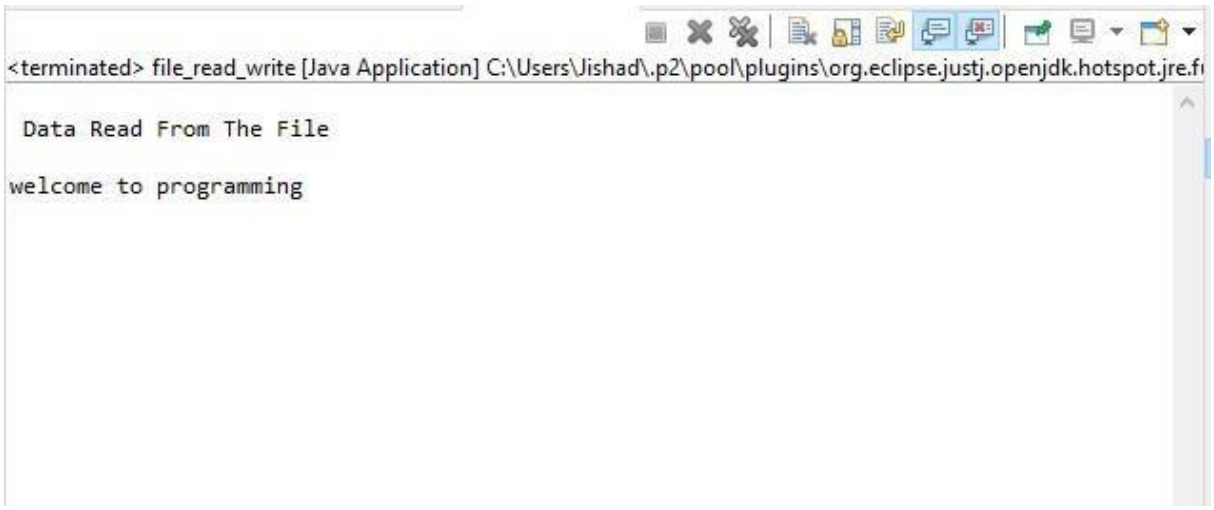
PROGRAM-44

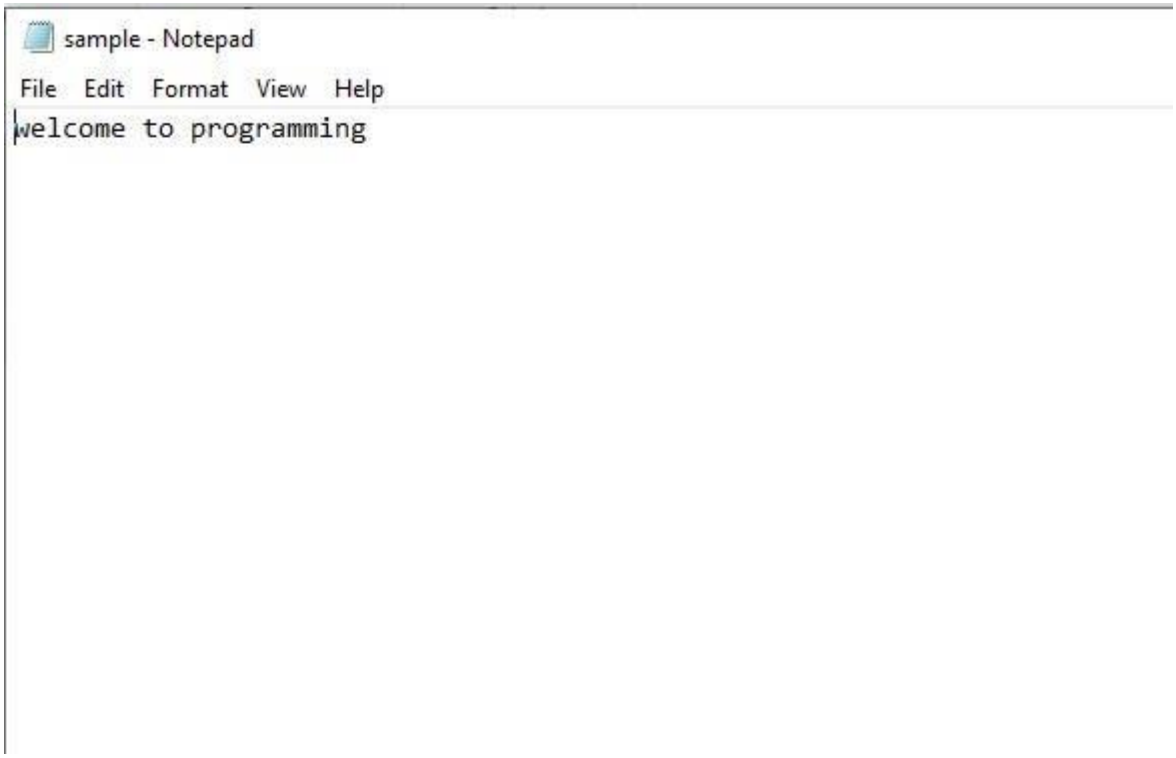
AIM: Write a program to write to a file, then read from the file and display the contents on the console.

PROGRAM CODE

```
import java.io.BufferedReader;
import java.io.FileReader;
 import java.io.FileWriter;
 import java.io.IOException;
 public class file_read_write {
public static void main(String[] args)
        {try
 {
        FileWriter fw=new
FileWriter("C:\\Users\\Jishad\\Documents\\java\\cycle6\\sa
mple",true);
        fw.write("welcome to
        programming");fw.close();
         FileReader reader =
                        new
FileReader("C:\\Users\\Jishad\\Documents\\java\\cycle6\\
 sa mple");
BufferedReader b= new BufferedReader(reader);
String line;
System.out.println("\n Data Read From The File \n");while ((line
= b.readLine()) != null) {
System.out.println(line);
}
reader.close();
}catch (IOException e)
 {
System.out.println("\n Error Occured...");
}}}
```

OUTPUT:





RESULT:

The above program is successfully executed and obtained the output

PROGRAM-45

AIM: Write a program to copy one file to another

PROGRAM CODE

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

public class copy_file {

   public static void  main(String args[]) throws Exception {
                        Scanner sc= new Scanner(System.in);
                        System.out.println("enter the first file:");
                        String file1=sc.next();
                        System.out.println("enter the Second file:");
                        String file2 =sc.next();
                        sc.close();
                         FileReader fin = new FileReader(file1);
                FileWriter fout = new FileWriter(file2, true);
                         int c;
                       while ((c = fin.read()) != -1) {
                         fout.write(c);
                          }
                         System.out.println("copy file1 to file2 ");
                        fin.close();
                        fout.close();
                 }}
```

OUTPUT:

```
PS C:\Users\USER\Desktop\Qs3> javac copy_file.java
PS C:\Users\USER\Desktop\Qs3> java copy_file
enter the first file:
file1.txt
enter the Second file:
file2.txt
copy file1 to file2
PS C:\Users\USER\Desktop\Qs3>
```

RESULT:

The above program is successfully executed and obtained the output

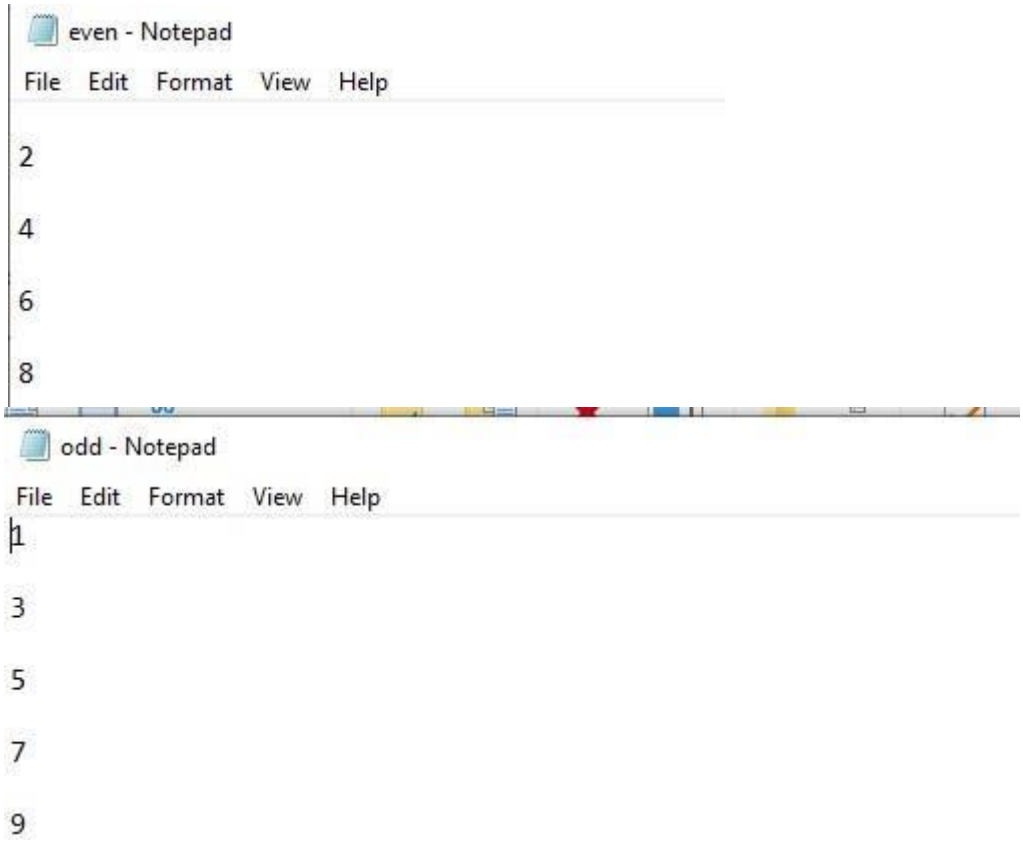
PROGRAM-46

AIM: Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

PROGRAM CODE

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Scanner;
public class copy_evnod {
public static void main(String args[]) throws IOException
{
FileInputStream fr = new
FileInputStream("C:\\Users\\Jishad\\Documents\\java\\cycle6\\in
teg er.txt");
FileOutputStream fw1 = new
FileOutputStream("C:\\Users\\Jishad\\Documents\\java\\cycle6\\
even.txt");
FileOutputStream fw2 = new
FileOutputStream("C:\\Users\\Jishad\\Documents\\java\\cycle6\\o
dd.txt");
System.out.println("\nFrom the file 'integers.txt' ,Odd Numbers
are copied to 'odd.txt'file and Even numbers are copied to
'even.txt' file\n");
int i;
while((i=fr.read()) != -1)
{
if(i%2==0)
fw1.write(i);
Else
fw2.write(i);
}
fr.close();
fw1.close();
fw2.close();
}
}
```

OUTPUT:



RESULT:

The above program is successfully executed and obtained the output

PROGRAM-47

AIM: Client server communication using Socket – TCP/IP

PROGRAM CODE

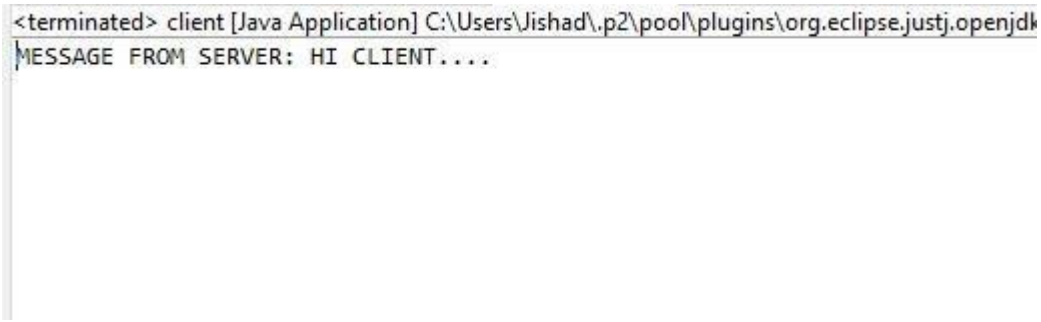
Client.java

```
import java.net.*;
import java.io.*;
public class client{
public static void main(String args[])
throws Exception{
try {
Socket sk = new Socket
("localhost", 1234);
PrintWriter pw = newPrintWriter(sk.getOutputStream(), true);
pw.println("HELLOSERVER ..!!!!!");
InputStreamReader isr= new InputStreamReader(sk.getInputStream());
BufferedReader br =new BufferedReader(isr);
String str=br.readLine();
System.out.println("MESSAGE FROM SERVER: "+str);
pw.close();
sk.close();
}
catch(Exception e) {
System.out.println("An error occured.." +e);
}}}
```

Server.java

```
import java.net.*;
import java.io.*;
public class server {
public static void main(String[] args) throws Exception {
try {
ServerSocket ss = new ServerSocket(1234);
System.out.println("SERVER IS WAITING FOR THE CLIENT.    ");
Socket sk = ss.accept();
System.out.println("CONNECTION ESTABLISHED !!!");
InputStreamReader isr= new InputStreamReader(sk.getInputStream());
BufferedReader br =new BufferedReader(isr);
String str =br.readLine();
System.out.println("MESSAGE FROM CLIENT: "+str);
PrintWriter pw = new PrintWriter(sk.getOutputStream(), true);
pw.println("HI
CLIENT.        ");
pw.close();
}
catch(Exception e) {
System.out.println("An
error occured.."+e);
}}}
```

OUTPUT:



RESULT:

The above program is successfully executed and obtained the output

PROGRAM-48

AIM: Client Server communication using DatagramSocket - UDP

PROGRAM CODE

Client.java

```
import java.io.*;
import
java.net.*;
public class client_udp {
        public static void main(String[] args) throws
            IOException {DatagramSocket client= new
            DatagramSocket(); InetAddress
            add=InetAddress.getByName("localhost");
            String str ="Hello...Server";
            byte[] bufBytes =
            str.getBytes(); DatagramPacket
            datagramPacket=new
DatagramPacket(bufBytes,bufBytes.length,ad
            d,1234);
            client.send(datagramPacket);
            client.close();
          }
}
```

Server.java

```
import java.io.*;
import java.net.*;
public class server_udp {
        public static void main(String[] args) throws
    IOException {DatagramSocket server=new
    DatagramSocket(1234);
    byte[] buf=new byte[256];
    DatagramPacket packet=new
    DatagramPacket(buf,buf.length);
    server.receive(packet);
    String reply =new
    String(packet.getData());
    System.out.println("\n Client Says :
    "+reply);server.close();
  }

}
```

OUTPUT:

RESULT:

The above program is successfully executed and obtained the output