# OBJECT ORIENTED PROGRAMMING LAB

**SUBMITTED BY,**

**BLESSY P ROY**

**S2 MCA**

**ROLL NO:MCA214**

# LABCYCLE 3

## PROGRAM:1

**AIM:** Area of different shapes using overloaded functions

**ALGORITHM**

STEP 1: Create a class called area

STEP 2: Create 3 member functions to calculate the area of square, rectangle and circle

STEP 3: Perform the area finding operations inside the functions

STEP 4: create object of the class area

STEP 5: call the functions which is created using objects

STEP 6: print the values of area of each shape

STEP 7: Stop

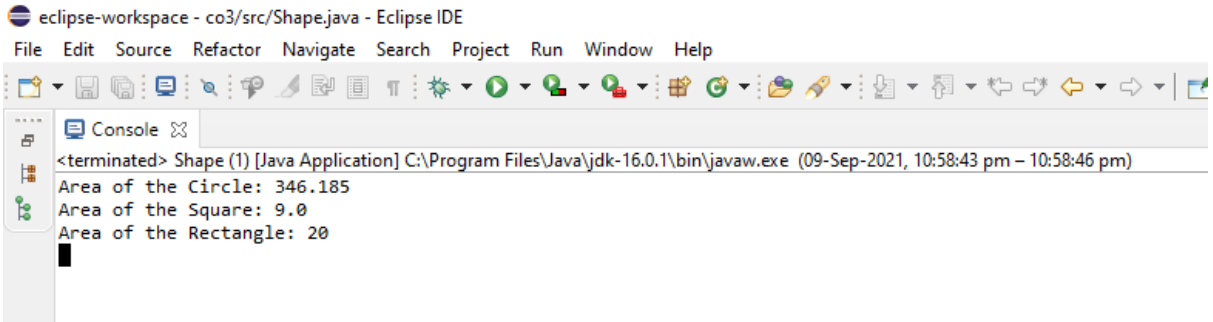
**SOURCE CODE**

| | |
|---|---|
| Shape.java | class overload<br>{<br>void Area(float a)<br>{<br>float A = a * a;<br>System.out.println("Area of the Square: " + A);<br>}<br><br>void Area(double a)<br>{<br>double A = 3.14 * a * a;<br>System.out.println("Area of the Circle: " + A);<br>}<br><br>void Area(int a, int b)<br>{<br>int A = a * b;<br>System.out.println("Area of the Rectangle: " + A);<br>}<br>}<br>class Shape {<br>public static void main(String[] args)<br>{<br>overload obj = new overload();<br>obj.Area(10.5);<br>obj.Area(3);<br>obj.Area(5, 4);<br>}<br>} |

**RESULT**

The above program  is executed and obtained the output.

**OUTPUT**



## PROGRAM: 2

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

**ALGORITHM**

STEP 1: Create a class employee

STEP 2: Create data members empid, name salary, address

STEP 3: Create a constructor to initialize these data members

STEP 4: Create another class teacher which is inherited from the class employee

STEP 5: Create teacher class's data members and initialize it with constructor

STEP 6: Create function to display all data members

STEP 7: Create array of objects

STEP 8: call the display function to print all the data members

STEP 9: Stop

**SOURCE CODE**

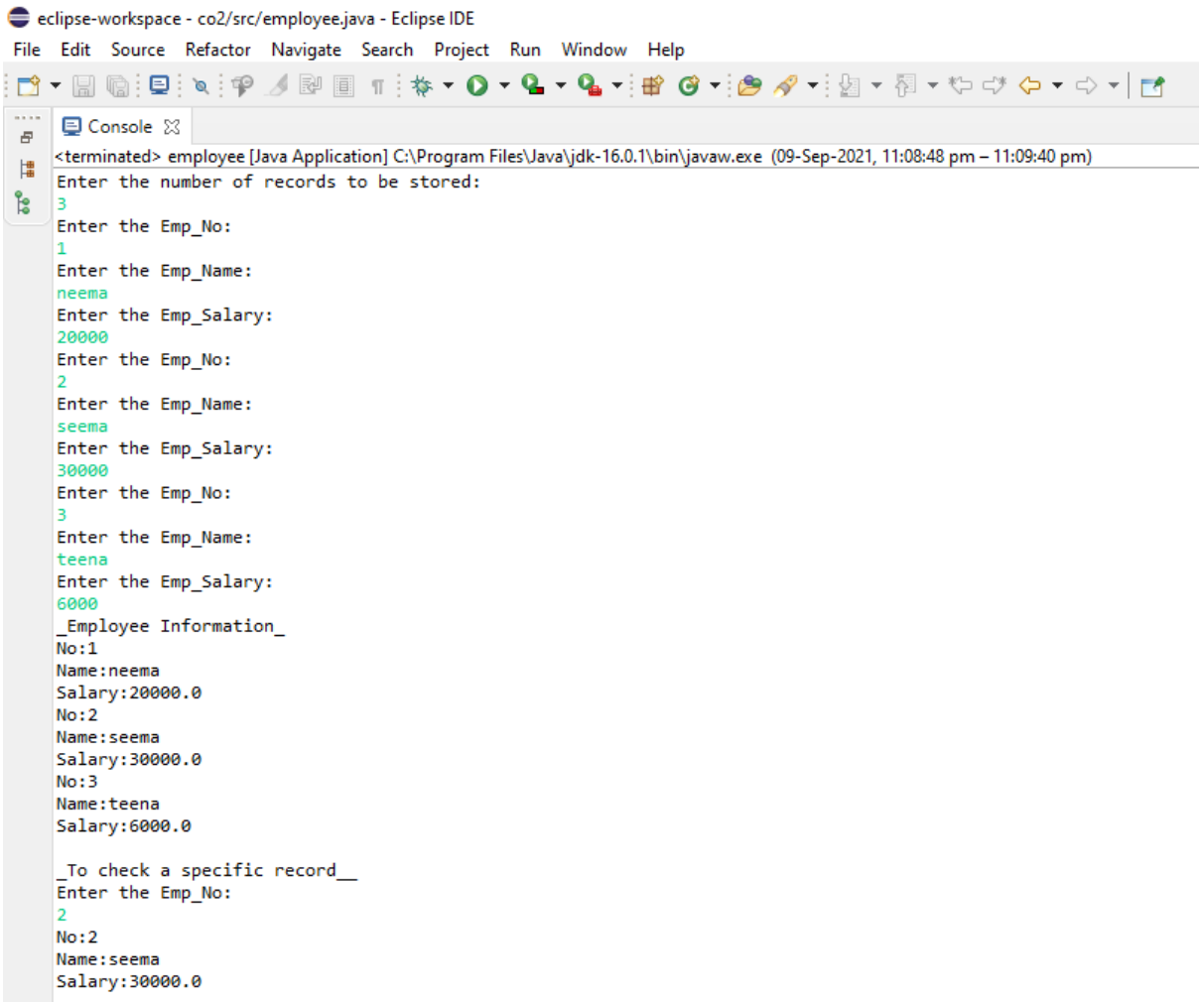
| | |
|---|---|
| Employee.java | import java.util.Scanner;<br>public class Employee {<br>int empId;<br>double salary;<br>String name,address;<br>public Employee(int empId,double salary,String name,String address)<br>{<br>this.empId=empId;<br>this.salary=salary;<br>this.name=name;<br>this.address=address;<br>}<br>}<br><br>class Teacher extends Employee<br>{<br>String department,subject;<br>public Teacher(int empId,double salary,String name,String address,String department,String subject)<br>{ |

```
    super(empId,salary,name,address);
    this.department=department;
    this.subject=subject;
}
void display()
{

System.out.println("\nEMPID\t\tSALARY\t\tNAME\t\tADDRESS\t\t
DEPARTMENT\t\tSUBJECT\n");

System.out.println(empId+"\t\t"+salary+"\t\t"+name+"\t\t"+address+"\
t\t"+department+"\t\t"+subject+"\n");
}


    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the total number of records to be stored
: ");
        int count=s.nextInt();
        Teacher t[]=new Teacher[count];
        int i;
        for(i=0;i<count;i++)
        {
            System.out.println("Enter
empid,salary,name,address,department and subject");
            int p=s.nextInt();
            double a=s.nextDouble();
            String b=s.nextLine();
            String c=s.nextLine();
            String d=s.nextLine();
            String e=s.nextLine();
            t[i]=new Teacher(p,a,b,c,d,e);

        }
        for(i=0;i<count;i++)
        {
            t[i].display();
        }

    }


}
```

## RESULT

The above program  is executed and obtained the output.

## OUTPUT



## PROGRAM: 3

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

**ALGORITHM**

STEP 1: Create a class person with data members name, gender, address, age

STEP 2: Create constructor to initialize these object

STEP 3: Create another class Employee that inherited the properties of class person and also creates its data members like empid, companyname , qualification, salary and its constructor

STEP 4: Create another class Teacher that inherits the properties of class employee and creates its data members

STEP 5: Creates array of object

STEP 6: Using these arrays of objects print the values

STEP 7: Stop

**SOURCE CODE**

| | |
|---|---|
| person.java | import java.util.Scanner;<br>public class person {<br>String Name;<br>String Address;<br>String Gender;<br>int Age;<br><br>public person(String nam,String gen, String adr, int age) {<br>Name=nam;<br>Gender=gen;<br>Address=adr;<br>Age=age;<br>}<br>static class Employee extends person<br>{<br>int Empid;<br>String company_name;<br>String Qualification;<br>double Salary;<br><br>public Employee(String nam,String gen,<br>String adr, int age,int id,String cname,String qu, double sal) |

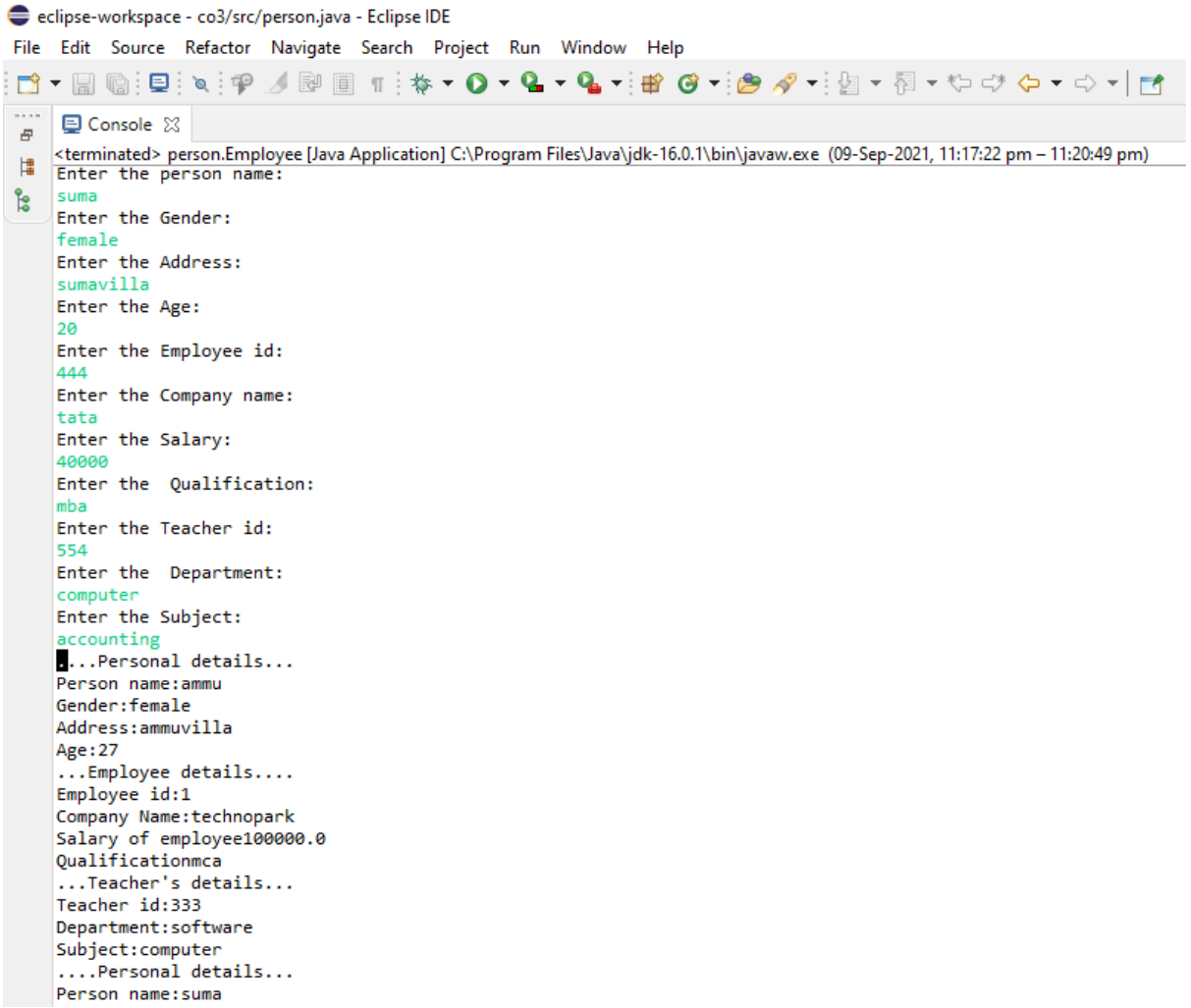
```
{
        super(nam,gen,adr,age);
        Empid=id;
        company_name=cname;
        Qualification=qu;
        Salary=sal;

}
static class Teacher extends Employee
{
        int teacher_id;
        String department;
        String subject;
public Teacher(String nam,String gen, String adr, int age,int
id,String cname,String qu,
    double sal,int tid,String dept,String sub) {
        super(nam,gen,adr,age,id,cname,qu,sal);
            teacher_id=tid;
            department=dept;
            subject=sub;
            }
        void display()
        {
        System.out.println("....Personal details...");
        System.out.println("Person name:"+Name);

System.out.println("Gender:"+Gender);

System.out.println("Address:"+Address);
        System.out.println("Age:"+Age);
        System.out.println("...Employee details....");
        System.out.println("Employee id:"+Empid);
    System.out.println("Company Name:"+company_name);
        System.out.println("Salary of employee"+Salary);

System.out.println("Qualification"+Qualification);
    System.out.println("...Teacher's details...");
    System.out.println("Teacher id:"+teacher_id);
    System.out.println("Department:"+department);
    System.out.println("Subject:"+subject);
            }}
    public static void main(String[] args) {

        int i,count;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter number of person:");
        count=s.nextInt();
```

```
                Teacher t[]=new Teacher[count];

                for(i=0;i<count;i++)
                {
                        System.out.println("Enter the person
name:");
                        String nam1=s.next();
                        System.out.println("Enter the Gender: ");
                        String gen1=s.next();
                        System.out.println("Enter the Address: ");
                        String adr1=s.next();
                        System.out.println("Enter the Age:");
                        int age1=s.nextInt();
                        System.out.println("Enter the Employee id:
");
                        int id1=s.nextInt();
                        System.out.println("Enter the Company
name: ");
                        String cname1=s.next();
                        System.out.println("Enter the Salary:");
                        double sal1=s.nextDouble();
                        System.out.println("Enter the
Qualification:");
                        String qu1=s.next();

                        System.out.println("Enter the Teacher id: ");
                        int tid1=s.nextInt();
                        System.out.println("Enter the
Department:");
                        String dept1=s.next();
                        System.out.println("Enter the Subject:");
                        String sub1=s.next();

                        t[i]=new
Teacher(nam1,gen1,adr1,age1,id1,cname1,qu1,sal1,tid1,dept1,sub1
);}
                for(i=0;i<count;i++)
                {
                        t[i].display();

                }
        }

}
}
```

## RESULT

The above program  is executed and obtained the output.

## OUTPUT



## PROGRAM: 4

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

**ALGORITHM**

STEP 1: Create a class publisher, Book, Literature and Fiction using inheritance

STEP 2: Create each class's data members and member functions

STEP 3: Read this information from the user

STEP 4: Print the details of book from the category

STEP 5: Stop

**SOURCE CODE**

| | |
|---|---|
| Publisher.java | import java.util.Scanner;<br>public class Publisher {<br>String Pubname;<br><br>Publisher()<br>{<br>Scanner s=new Scanner(System.in);<br>System.out.println("Enter publisher name");<br>Pubname=s.next();<br>}<br>}<br>class Book extends Publisher<br>{<br>String title, author;<br>int price;<br><br>Book()<br>{<br>Scanner s=new Scanner(System.in);<br>System.out.println("Enter Title of the book");<br>title=s.next();<br>System.out.println("Enter Author's name");<br>author=s.next();<br>System.out.println("Enter price");<br>price=s.nextInt();<br>}<br>}<br>class Literature extends Book<br>{<br>Literature()<br>{<br>System.out.println("Literature Books");<br>} |

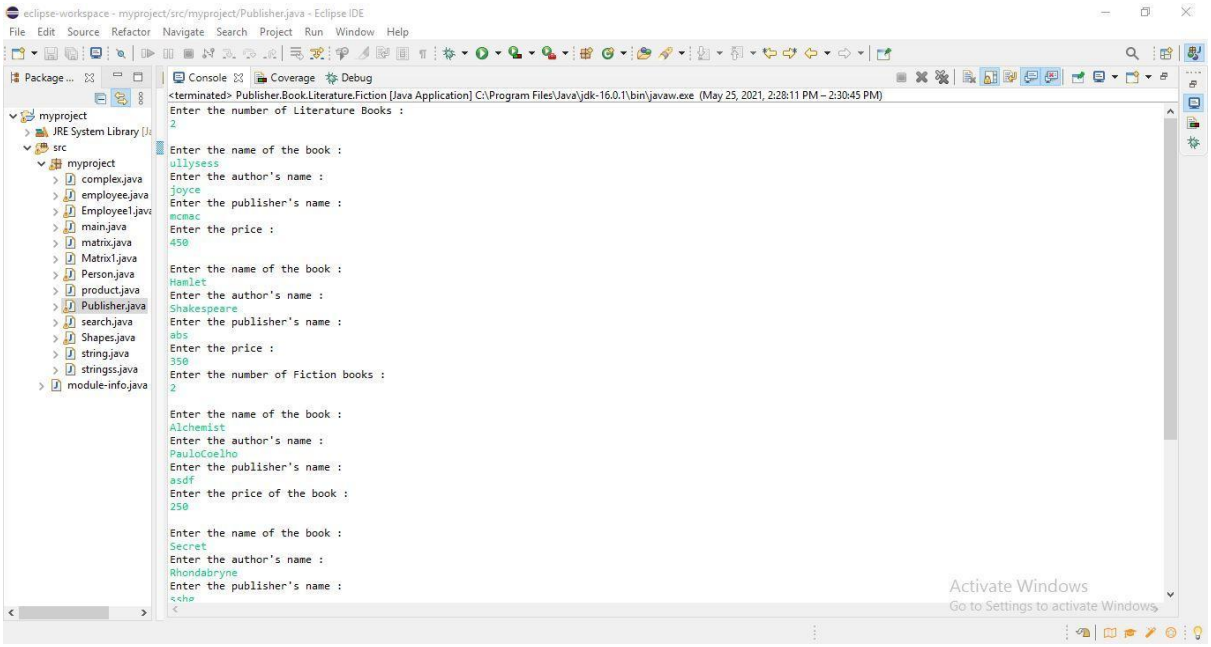
```
 void display()
 {
        System.out.println("Publisher name: "+Pubname);
        System.out.println("Title of the book: "+title);
        System.out.println("Author's name: "+author);
        System.out.println("Price: "+price);
 }

}
class Fiction extends Literature
{
Fiction()
{
        System.out.println("Friction Books");
}
void display()
{
        super.display();
}

public static void main(String args[])
        {
                int n;
                Scanner s=new Scanner(System.in);

                System.out.println("Enter the No of literature book: ");
                int a=s.nextInt();
                Literature L[]=new Literature[a];
                for(int i=0;i<a;i++)
                {
                        L[i]=new Literature();
                }

                System.out.println("Enter the No of Fiction book: ");
                int b=s.nextInt();
                Fiction F[]=new Fiction[b];
                for(int i=0;i<b;i++)
                {
                        F[i]=new Fiction();
                }
                int no;
                System.out.println("Enter your choice of book");
                no=s.nextInt();
                int type =no;
                switch (no)
                {
                case 1:
```

```
                        System.out.println(".....Details of literature
books");
                        for(int i=0;i<a;i++)
                        L[i].display();
                        break;
                case 2:
                        System.out.println(".....Details of fiction
books");
                        for(int i=0;i<b;i++)
                        F[i].display();
                        break;
                default:
                        System.out.println("Wrong input");
                }
        }
}
```

**RESULT**

The above program  is executed and obtained the output.

**OUTPUT**



## PROGRAM: 5

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

**ALGORITHM**

STEP 1: Create classes student and sports

STEP 2: Create another class Result inherited from Student and Sports

STEP 3: Create a function called display to print

STEP 4: display academic and sports score of student

STEP 5: Stop

**SOURCE CODE**

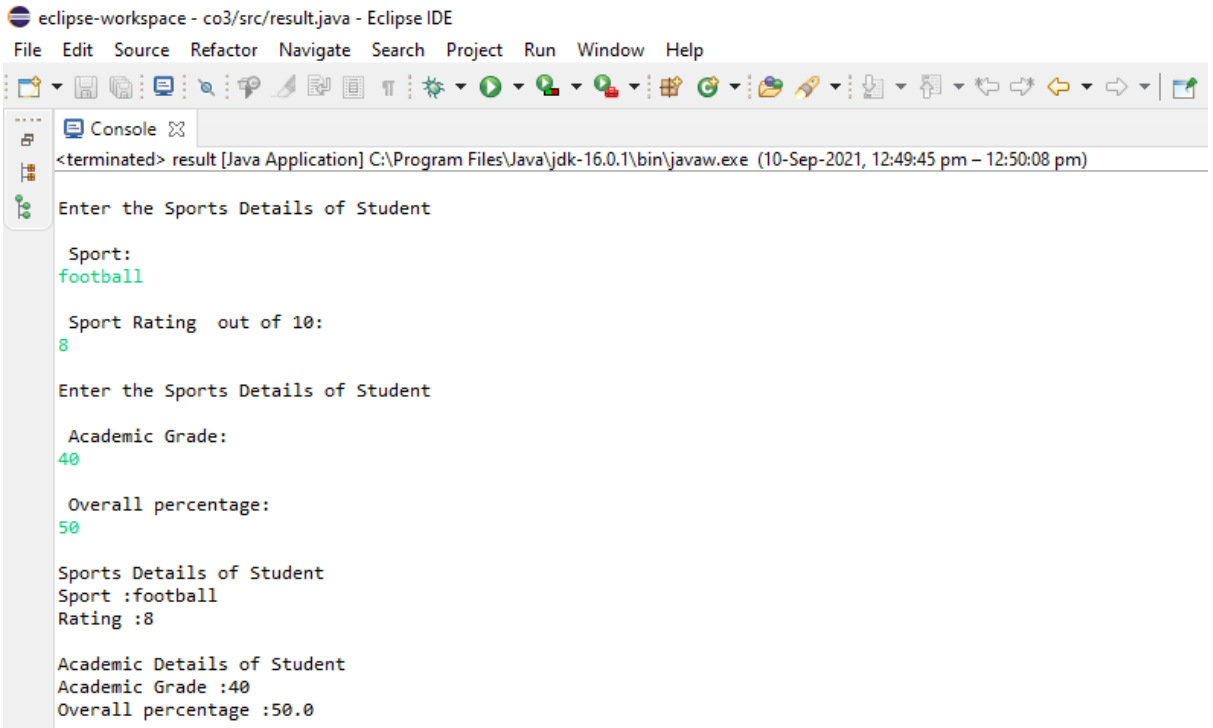
| | |
|---|---|
| result.java | import java.util.Scanner;<br>class sports{<br>String sport;<br>int Rating;<br>sports(String spo, int ra){<br>sport = spo;<br>Rating = ra;<br>}<br><br>}<br>class student extends sports{<br>String Grade;<br>double Overall_per;<br>student(String spo, int ra,String gd, double per ){<br>super(spo, ra);<br>Grade = gd;<br>Overall_per = per;<br>}<br>}<br>public class result extends student {<br>result(String spo, int ra,String gd, double per ){<br>super(spo, ra, gd, per);<br>}<br>void display(){<br>System.out.println("\nSports Details of Student");<br>System.out.println("Sport :"+sport);<br>System.out.println("Rating :"+Rating);<br>System.out.println("\nAcademic Details of Student");<br>System.out.println("Academic Grade :"+Grade);<br>System.out.println("Overall percentage :"+Overall_per);<br>}<br><br>public static void main(String[] args) { |

```
        Scanner sc =new Scanner(System.in);
        System.out.println("\nEnter the Sports Details of Student");
        System.out.println("\n Sport: ");
        String a =sc.next();
        System.out.println("\n Sport Rating  out of 10: ");
        int b =sc.nextInt();
        System.out.println("\nEnter the Sports Details of Student");
        System.out.println("\n Academic Grade: ");
        String c =sc.next();
        System.out.println("\n Overall percentage: ");
        double d =sc.nextDouble();
        sc.close();
        result obj= new result(a,b,c,d);
        obj.display();
    }
}
```

## RESULT

The above program  is executed and obtained the output.

## OUTPUT



## PROGRAM: 6

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

**ALGORITHM**

STEP 1: Create an interface having prototypes of function area() and perimeter()

STEP 2: Create classes that implements the interface

STEP 3: Create two classes circle and rectangle

STEP 4: find the perimeter and area of rectangle and circle

STEP 5: Print the results

STEP 6: Stop

**SOURCE CODE**

| | |
|---|---|
| calculate.java | import java.util.Scanner;<br>interface math{<br>void area();<br>void perimeter();<br>void input();<br>}<br>class circle implements math{<br>float r;<br>double pi=3.14,area,perimeter;<br>Scanner cr=new Scanner(System.in);<br><br>public void input() {<br>System.out.println("enter the radius:");<br>r=cr.nextFloat();<br>}<br><br>@Override<br>public void area() {<br>area=pi*r*r;<br>System.out.println("Area of the circle:"+area);<br><br>}<br><br>@Override<br>public void perimeter() {<br>perimeter=2*pi*r;<br>System.out.println("Perimeter of the circle:"+perimeter);<br>}<br>} |

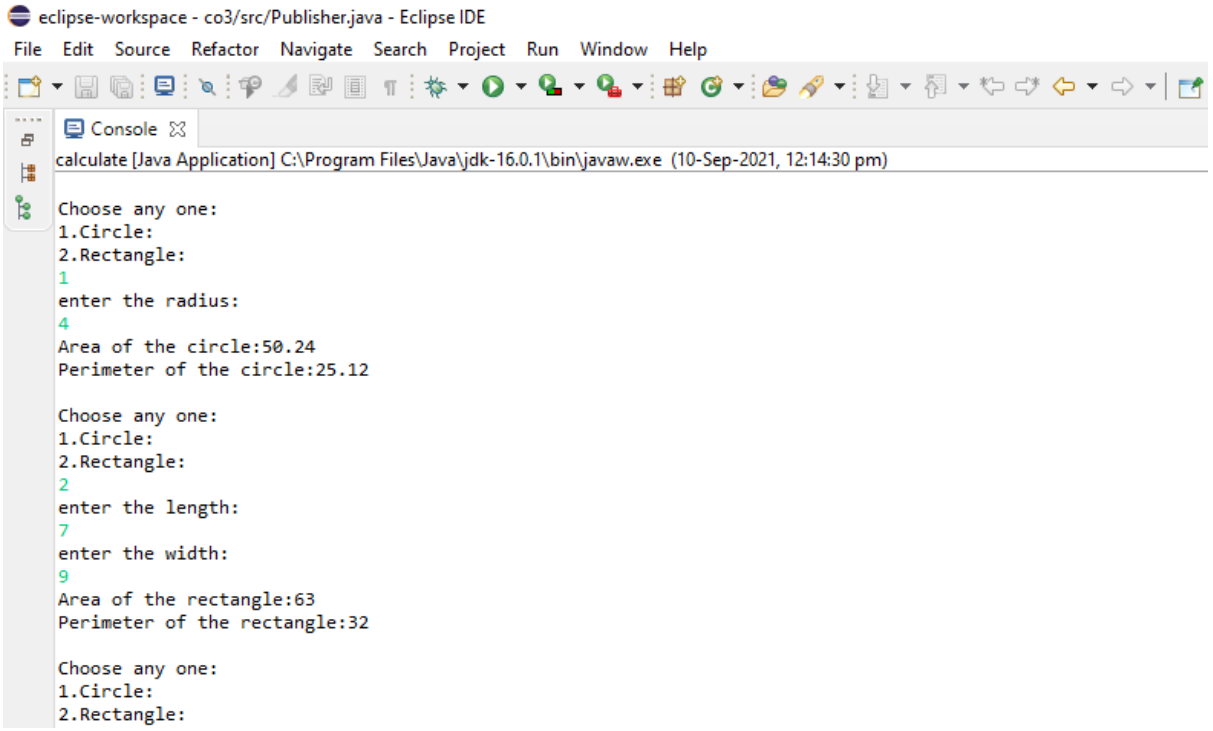
```
class rectangle implements math{
	int l,w,area,perimeter;
	Scanner rl=new Scanner(System.in);

	@Override
	public void input() {
		System.out.println("enter the length:");
		l=rl.nextInt();
		System.out.println("enter the width:");
		w=rl.nextInt();

	}

	@Override
	public void area() {
		area=l*w;
		System.out.println("Area of the rectangle:"+area);

	}

	@Override
	public void perimeter() {
		perimeter=2*(l+w);
		System.out.println("Perimeter of the
rectangle:"+perimeter);

	}

}
public class calculate{

	public static void main(String[] args) {
		circle obj1=new circle();
		rectangle obj2=new rectangle();
		int y=0;
		while(y<3) {
		Scanner sc=new Scanner(System.in);
		System.out.println("\nChoose any one:");
	   System.out.println("1.Circle:\n2.Rectangle:");
		y=sc.nextInt();
		   switch(y) {
			case 1:
				obj1.input();
				obj1.area();
				obj1.perimeter();
				break;
			case 2:
```

```
                                        obj2.input();
                                        obj2.area();
                                        obj2.perimeter();
                                        break;
                                default:
                                        System.out.println("wrong
choice!!!!");
                                        }
                        }
                }
}
```

**RESULT**

The above program  is executed and obtained the output.

**OUTPUT**



## PROGRAM: 7

**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

**ALGORITHM**

STEP 1: Create a class Co37.java

STEP 2: Create another class called cal that inherits from the interface cal_bill

STEP 3: Create a display function

STEP 4: Print the values as shown as the question

STEP 5: Stop

**SOURCE CODE**

co37.java

```
import java.util.*;
import java.time.*;

interface cal_bill
{
        public void calc();

}

class cal implements cal_bill
{
        int p_id,qty,up,tot,o_n,date;
        String name;
        public void details()
        {
                Scanner ss = new Scanner(System.in);

                System.out.println("Enter Product ID : ");
                p_id = ss.nextInt();
                System.out.println("Enter Name : ");
                name = ss.next();
```

```
                System.out.println("Enter Quantity : ");
                qty = ss.nextInt();
                System.out.println("Unit Price : ");
                up = ss.nextInt();

        }

        public void order()
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter Order No : ");
                o_n = sc.nextInt();
        }

        public void calc()
        {
                tot = qty * up;
                System.out.println("   "+p_id+"\t\t "+name+"\t
"+qty+"\t\t    "+up+"\t     "+tot);


        }

        void display()
        {
                System.out.print("\n");
                System.out.println("BILL");
                System.out.println("==========");
                System.out.print("\n");
                System.out.println("Order No : " + o_n);
                LocalDate obj = LocalDate.now();
                System.out.print("\n");
                System.out.println("Date : " + obj);
                System.out.print("\n");
                System.out.println("Product Id\tName\tQuantity\tunit
price\tTotal");

System.out.print("_____________________________________________
_____________________________");
                System.out.print("\n");

        }

}

public class co37
{
  public static void main(String[] args)
```

```
  {
        int no,net = 0;

        Scanner s = new Scanner(System.in);
        System.out.println("Enter no of products : ");
        no = s.nextInt();
        cal or = new cal();

        cal obj[] = new cal[no];

        or.order();

        for(int i=0;i<no;i++)
        {
              System.out.print("\n");
              System.out.println("Enter Details of product "+ (i+1));

System.out.println("==========================");

              obj[i] = new cal();
              obj[i].details();

        }


        or.display();

        for(int i=0;i<no;i++)
        {
              obj[i].calc();
              net = net + obj[i].tot;
        }


System.out.println("_____________________________________________
_____________________________");
        System.out.println("    "+"\t\t  "+"\t    "+"\t\tNet amount :"+"\t
"+net);


  }
}
```

## RESULT

The above program  is executed and obtained the output.

## OUTPUT

eclipse-workspace - co3/src/co37.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Console

```
<terminated> co37 [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (10-Sep-2021, 12:27:24 pm – 12:28:46 pm)
55

Enter Details of product 1
==========================
Enter Product ID :
565
Enter Name :
saree
Enter Quantity :
7000
Unit Price :
800

Enter Details of product 2
==========================
Enter Product ID :
99
Enter Name :
kurti
Enter Quantity :
90
Unit Price :
8000

BILL
==========

Order No : 55

Date : 2021-09-10

Product Id      Name    Quantity        unit price      Total
_____________________________________________________________
   565          saree     7000             800      5600000
   99           kurti     90               8000             720000
_____________________________________________________________
                                       Net amount :     6320000
```

# LABCYCLE 4

# PROGRAM 1

**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.

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a package contain functions to assign calculate and display .

STEP 3: Read inputs from user and assign values to objects.

STEP 4: Perform desired operations.

STEP 5: Print the Outputs.

STEP 6: Stop

**SOURCE CODE**

| | |
|---|---|
| area.java | package Graphics;<br>import java.util.Scanner;<br>interface figure{<br>void rectangle();<br>void triangle();<br>void square();<br>void circle();<br>}<br><br><br>public class area implements figure {<br>int x,y,b,h,r;<br>double a,ar,are,circle_ar;<br>Scanner sc=new Scanner(System.in);<br>public void rectangle()<br>{<br>System.out.println("Enter the length of Rectangle: ");<br>x=sc.nextInt();<br>System.out.println("Enter the breadth of Rectangle: ");<br>y=sc.nextInt();<br>a=x*y;<br>System.out.println("Area of Rectangle: "+a);<br><br>}<br>public void triangle()<br>{<br>System.out.println("Enter the base length of Triangle: "); |

```
                b=sc.nextInt();
                System.out.println("Enter the height of Triangle: ");
                h=sc.nextInt();
                ar=0.5*b*h;
                System.out.println("Area of Triangle: "+ar);
        }
        public void square()
        {
                System.out.println("Enter the side length of Square: ");
                a=sc.nextInt();
                are=a*a;
                System.out.println("Area of Square: "+are);
        }
        public void circle()
        {
                System.out.println("Enter the radius of Circle: ");
                r=sc.nextInt();
                circle_ar=Math.PI*r*r;
                System.out.println("Area of Circle: "+circle_ar);

        }
}
```

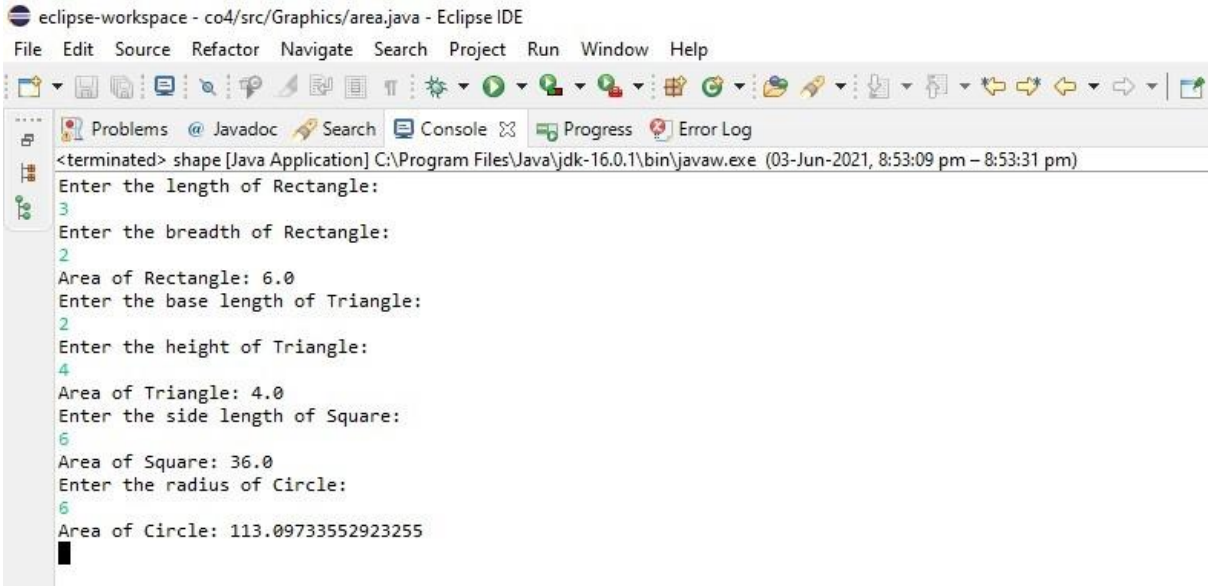
shape.java

```
package Graphics;
import Graphics.area;
public class shape
{

        public static void main(String[] args)
        {
                area ar=new area();
                ar.rectangle();
                ar.triangle();
                ar.square();
                ar.circle();
        }



}
```

**RESULT**

The above program  is executed and obtained the output.

**OUTPUT**



# PROGRAM 2

**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

## ALGORITHM

STEP 1 : Start

STEP 2: Define a package contain functions to assign calculate and display .

STEP 3: Read inputs from user and assign values to objects.

STEP 4: Perform desired operations.

STEP 5: Print the Outputs.

STEP 6: Stop

## SOURCE CODE

| | |
|---|---|
| math.java | package Arithmetic;<br>import Arithmetic.operands;<br>import java.util.Scanner;<br><br>public class math {<br><br>public static void main(String[] args) {<br>Scanner s = new Scanner(System.in);<br><br>float a,b;<br><br>System.out.println("_____ARITHMETIC OPERATIONS_____");<br><br>System.out.println("Enter any two numbers to be operated with:");<br>a = s.nextFloat();<br>b = s.nextFloat();<br><br>operands op = new operands(a,b);<br>op.add();<br>op.sub();<br>op.mul();<br>op.div();<br><br>s.close();<br>}<br>} |

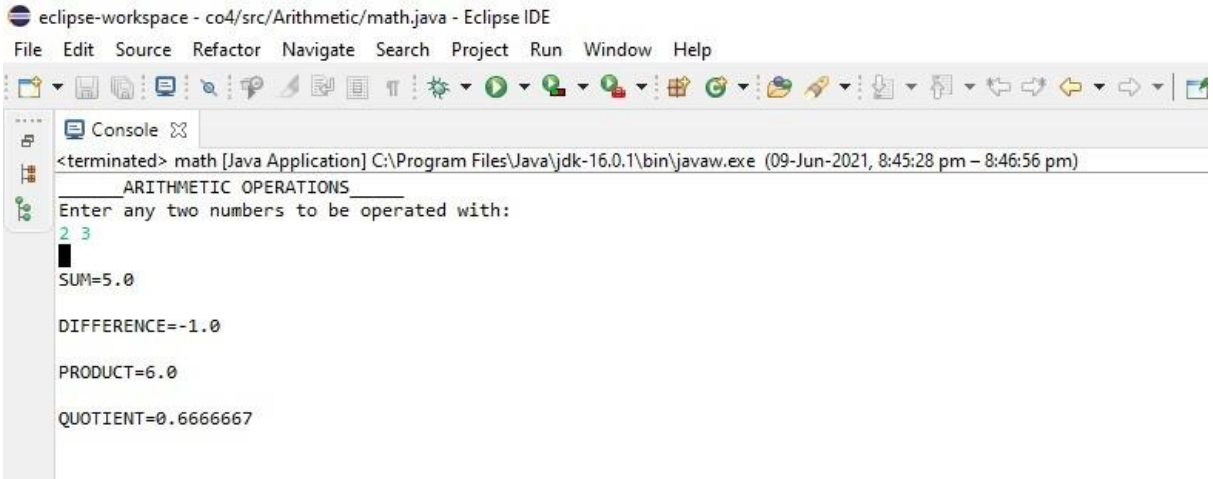
| | |
|---|---|
| operands.java | |

```
package Arithmetic;

interface operations{
	void add();
	void sub();
	void div();
	void mul();
}


public class operands implements operations {
	float x,y;

	public operands(float a, float b) {

		x=a;
		y=b;

	}

	public void add() {
		System.out.println("\nSUM="+(x+y));
	}

	public void sub() {
		System.out.println("\nDIFFERENCE="+(x-y));
	}

	public void div() {
		System.out.println("\nQUOTIENT="+(x/y));
	}

	public void mul() {
		System.out.println("\nPRODUCT="+(x*y));
	}


}
```

**RESULT**

The above program  is executed and obtained the output.

**OUTPUT**



# PROGRAM 3

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Read inputs as username and password.

STEP 3: Verify the username and password.

STEP 4: If its true; Print Succesful message:.

STEP 5: Else print error meessage.

STEP 6: Stop

**SOURCE CODE**

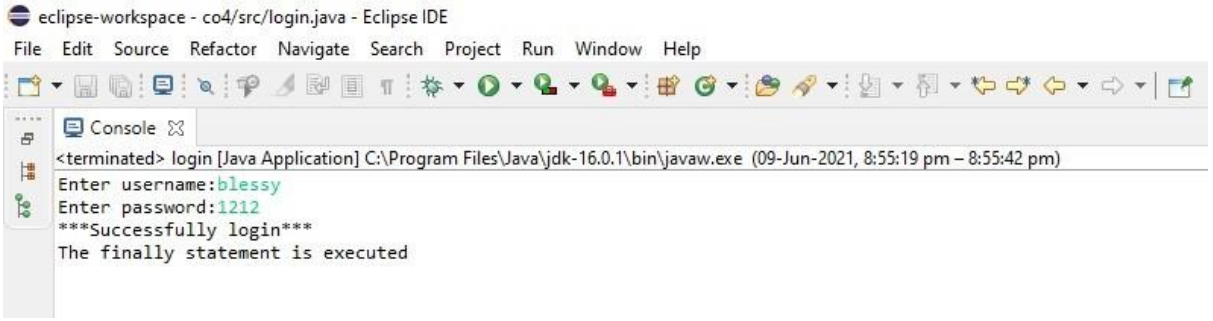
| | |
|---|---|
| login.java | import java.util.Scanner;<br>class Username_Exception extends Exception {<br>public Username_Exception(String string) {<br>super(string);<br>}<br>}<br><br>class Password_Exception extends Exception {<br>public Password_Exception(String string) {<br>super(string);<br>}<br>}<br>public class login {<br>public static void main(String[] args) {<br>Scanner sc = new Scanner(System.in);<br>String username, password;<br><br>System.out.print("Enter username:");<br>username = sc.nextLine();<br><br>System.out.print("Enter password:");<br>password = sc.nextLine();<br><br>int length = username.length();<br><br>try {<br>if(length>8)<br>throw new<br>Username_Exception("username must be atmost 8 characters!");<br>if(!password.equals("1212"))<br>throw new Password_Exception("Incorrect<br>password\nEnter the correct password??"); |

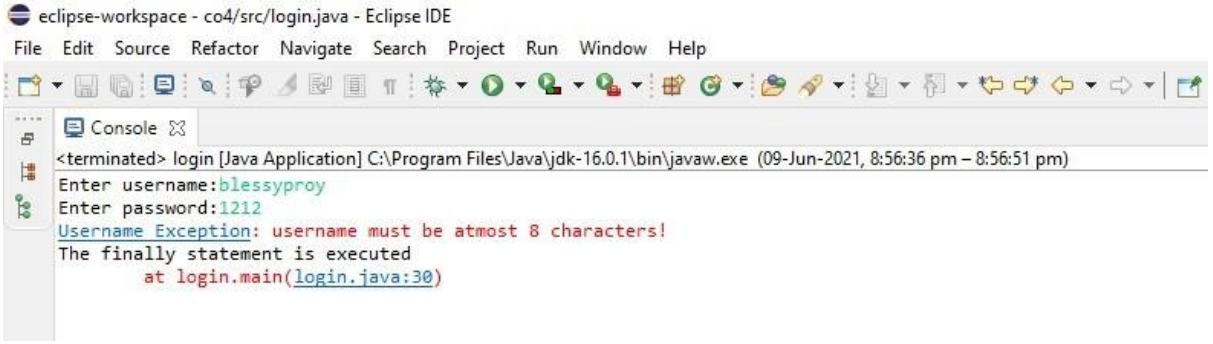
```
        else
            System.out.println("***Successfully login***");
        }
    catch (Username_Exception U) {
            U.printStackTrace();
    }
    catch (Password_Exception P) {
        P.printStackTrace();
    }
    finally {
    System.out.println("The finally statement is executed ");
            } }}
```

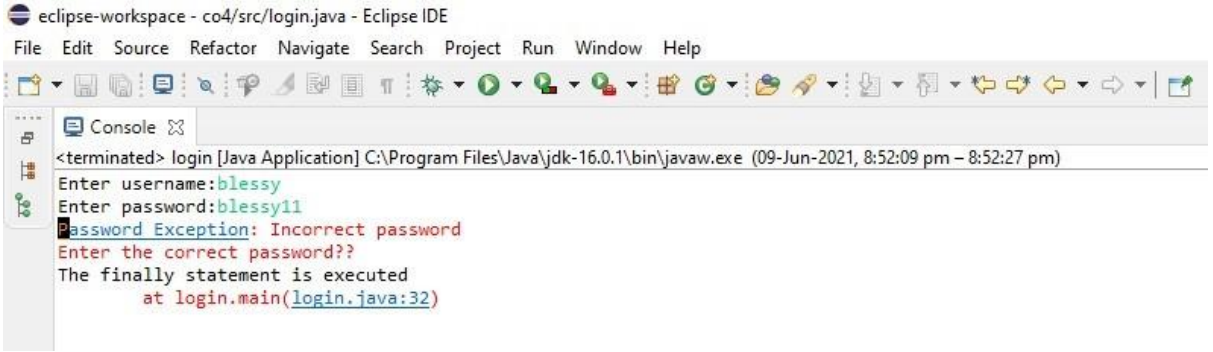
**RESULT**

The above program is executed and obtained the output.

**OUTPUT**



## PROGRAM: 4

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Enter a limit n.

STEP 3:Read n elements.

STEP 4:iterate the loop .

STEP 5:if i>0;. then

STEP 6: sum=sum+i

STEP 7: Else Print error message

STEP 8: Av=sum/n

STEP 9: Print av

STEP 10: Stop

**SOURCE CODE**

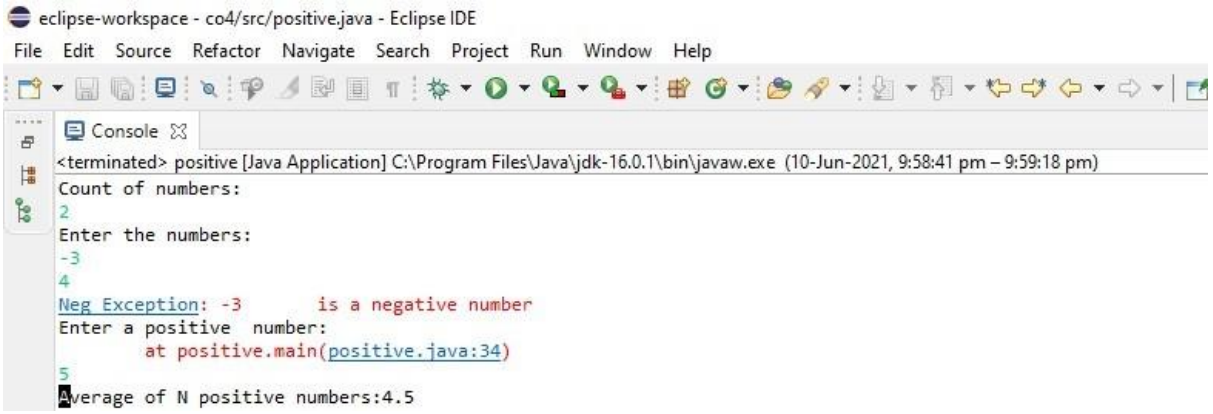
| | |
|---|---|
| positive.java | import java.util.*;<br><br>class Neg_Exception extends Exception<br>{<br>public Neg_Exception(String s)<br>{<br>super(s);<br>}<br>}<br>public class positive<br>{<br>public static void main(String[] args)<br>{<br>int i;<br>double sum=0;<br>Scanner sc=new Scanner(System.in);<br>System.out.println("Count of numbers: ");<br>int N=sc.nextInt();<br>int[] intArray=new int[N];<br>System.out.println("Enter the numbers: ");<br>for(i=0;i<N;i++)<br>{<br>intArray[i]=sc.nextInt(); |

```
                    }
                    for(i=0;i<N;i++)
                    {
                      try
                            {
                                if (intArray[i]<0)
                                      {
throw new Neg_Exception(+intArray[i]+"\tis a negative
number");
                                      }
                                      else
                                      {
                                            sum=sum+intArray[i];
                                      }

                            }
                            catch( Neg_Exception e)
                            {
                            e.printStackTrace();
                            System.out.println("Enter a positive
number: ");
                intArray[i]=sc.nextInt();
                sum=sum+intArray[i];
                    }}
                    System.out.println("Average of N positive
numbers:"+sum/N);
          }
}
```

**RESULT**

The above program is executed and obtained the output.

**OUTPUT**



# PROGRAM 5

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a class for multiplication and another for prime numbers.

STEP 3:Read a input as num.

STEP 4:Print the multiplication table of num.

STEP 5: Print num prime numbers.

STEP 6: Stop

**SOURCE CODE**

threadmul1.java

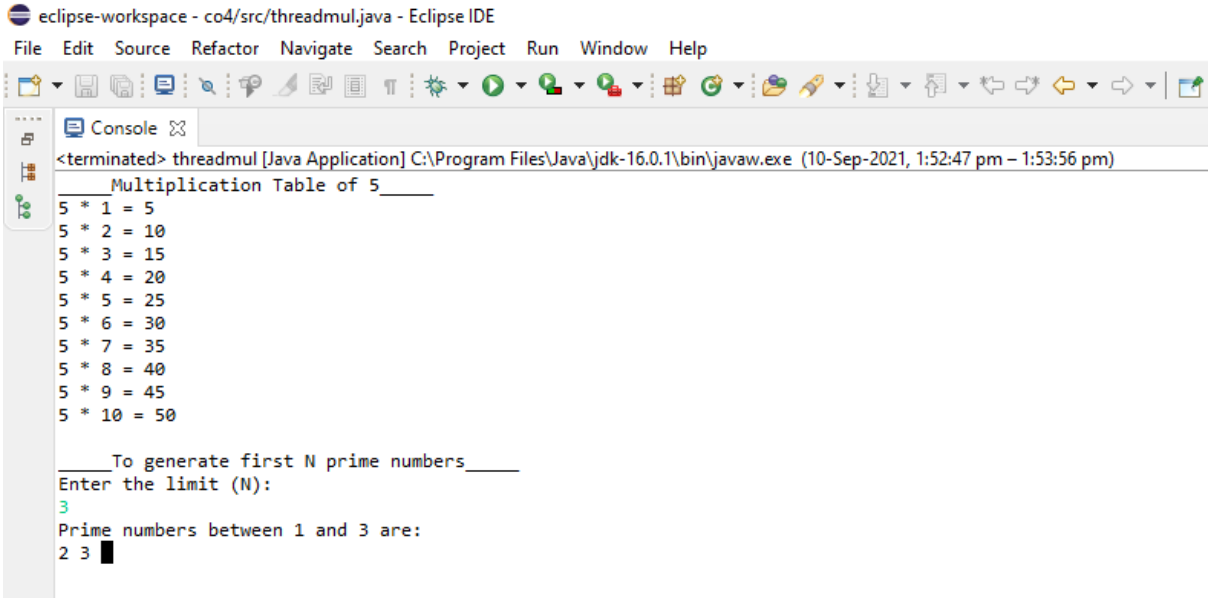
```
import java.util.Scanner;

class MulTable extends Thread{
  public void run() {
          int num = 5;
          System.out.printf("_____Multiplication Table of 5_____\n");
    for(int i = 1; i <= 10; ++i)
    {
          System.out.printf("%d * %d = %d \n", num, i, num * i);
    }
  }

}

class PrimeNo extends Thread{
          public void run() {
           int i, j,flag;
           Scanner s = new Scanner(System.in);
           System.out.println("\n_____To generate first N prime
numbers_____");
           System.out.println("Enter the limit (N):");
           int N = s.nextInt();


           System.out.println("Prime numbers between 1 and " + N + "
are:");

             for (i = 1; i <= N; i++)
             {
```

```
                if (i == 1 || i == 0)
                    continue;

                flag = 1;

                for (j = 2; j <= i / 2; ++j)
                {
                    if (i % j == 0)
                    {
                        flag = 0;
                        break;
                    }
                }

                if (flag == 1)
                    System.out.print(i + " ");
            }
        }
    }


public class threadmul {

        public static void main(String[] args) throws
InterruptedException {
                    MulTable m = new MulTable();
                    m.start();
                    m.sleep(100);

                    PrimeNo p = new PrimeNo();
                    p.start();
                    p.sleep(100);



        }

}
```

## RESULT

The above program  is executed and obtained the output.

## OUTPUT



# PROGRAM 6

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a class for Fibonacci and another for even numbers.

STEP 3:Read a input as n.

STEP 4:Print the Fibonnaci series of n number.

STEP 5: Printn n even numbers.

STEP 6: Stop

**SOURCE CODE**

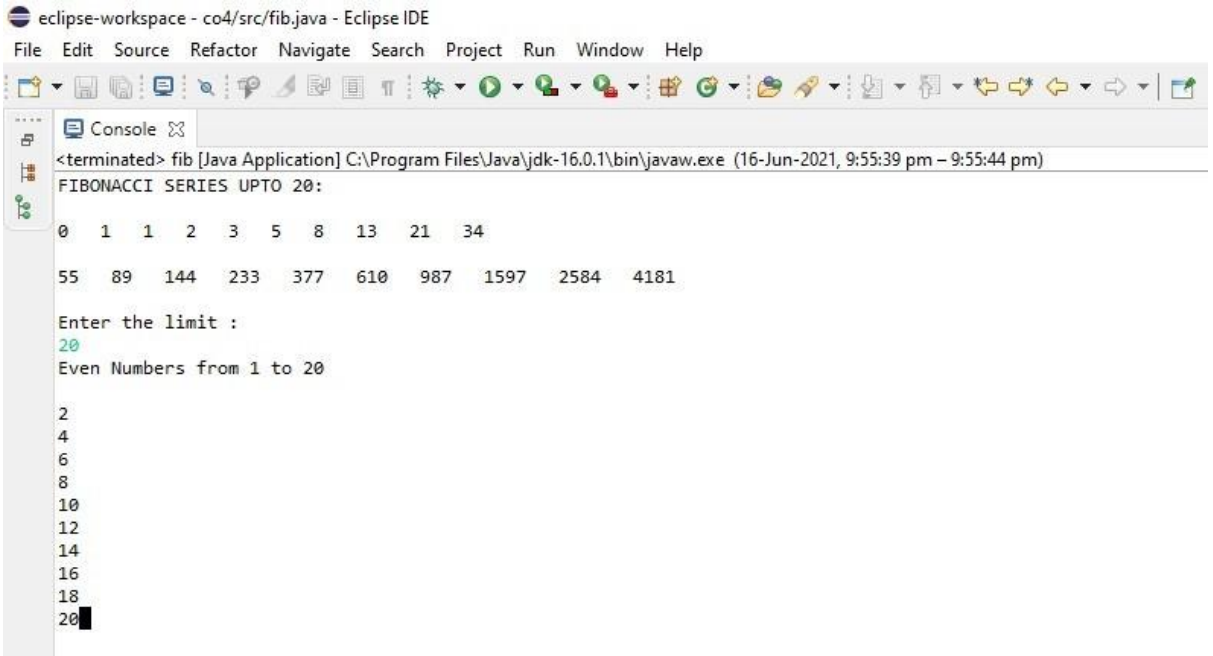
| | |
|---|---|
| fib.java | import java.util.Scanner;<br><br>class Fibonacci implements Runnable{<br>public void run(){<br>int a=0,b=1,c=0,l=20;<br>System.out.println("FIBONACCI SERIES UPTO "+l+":<br>\n");<br>while (l>0)<br>{<br>System.out.print(c+"  ");<br>a=b;<br>b=c;<br>c=a+b;<br>l=l-1;<br>if(l%10==0)<br>{<br>System.out.println("\n");<br>}<br>}<br><br>}<br>}<br><br>class EvenNumber implements Runnable{<br>public void run(){<br>int n;<br>Scanner sc=new Scanner(System.in);<br>System.out.println("Enter the limit : ");<br>n=sc.nextInt();<br>System.out.println("Even Numbers from 1 to "+n+"\n");<br>for(int i=1;i<=n;i++) { |

```
                        if(i%2==0) {
                                System.out.println(i);
                                }
                        }
                }
}
public class fib {

        public static void main(String[] args) {
                Fibonacci obj1=new Fibonacci();
                Thread t1=new Thread(obj1);
                t1.start();

                EvenNumber obj2=new EvenNumber();
                Thread t2=new Thread(obj2);
                t2.start();
        }

}
```

**RESULT**

The above program is executed and obtained the output.

**OUTPUT**



# PROGRAM 7

**AIM:** Producer/Consumer using ITC

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a class name as producer with members pname,pcode and price.

STEP 3: Define objects to Class and add 3 products and values to each data using the object.

STEP 4: Check whether the product has the lowest price using an if-else statement.

STEP 5: Print the details of the product.

STEP 6: Stop

**SOURCE CODE**

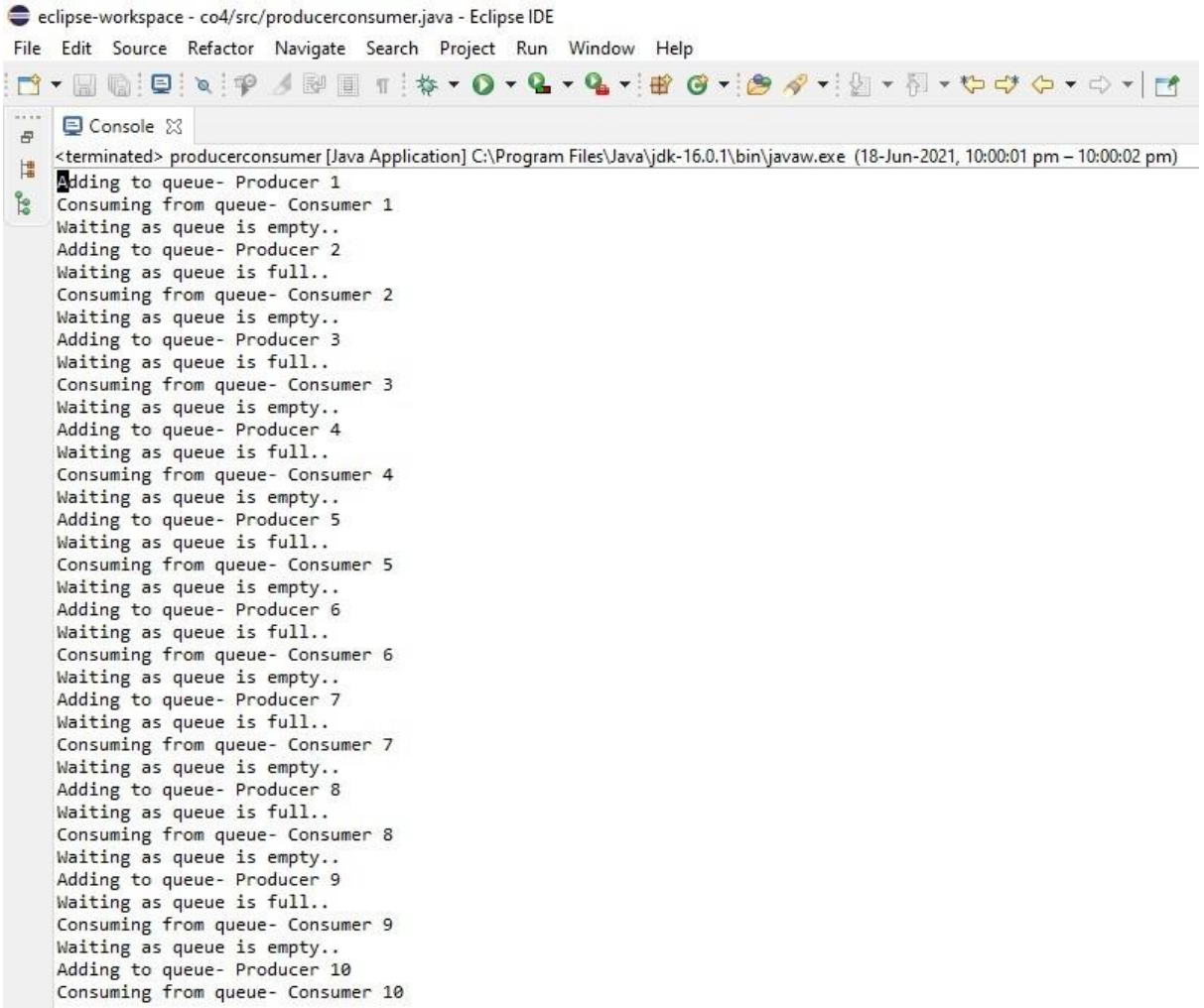
| | |
|---|---|
| producercustomer.java | import java.util.LinkedList;<br>public class producerconsumer {<br>public static void main(String[] args) {<br>// shared list<br>LinkedList<Integer> list = new LinkedList<Integer>();<br>Thread t1 = new Thread(new Producer(list), "Producer");<br>Thread t2 = new Thread(new Consumer(list), "Consumer");<br>t1.start();<br>t2.start();<br>}<br>}<br><br>//Producer task<br><br>class Producer implements Runnable{<br>LinkedList<Integer> list;<br>Producer(LinkedList<Integer> list){<br>this.list = list;<br>}<br>@Override<br>public void run() {<br>for(int i = 1; i <=10; i++){<br>synchronized(list) {<br>// If there is already an element in the list wait<br>while(list.size() >= 1){<br>System.out.println("Waiting as queue is full..");<br>try {<br>list.wait();<br>} catch (InterruptedException e) {<br>e.printStackTrace(); |

```
      }
    }
    System.out.println("Adding to queue- " +
Thread.currentThread().getName() + " " + i);
    list.add(i);
    list.notify();
  }
 }
}
}
//Consumer task
class Consumer implements Runnable{
LinkedList<Integer> list;
Consumer(LinkedList<Integer> list){
 this.list = list;
}
@Override
public void run() {
 for(int i = 1; i <= 10; i++){
  synchronized(list) {
    // if there is no element in the list wait
    while(list.size() < 1){
      System.out.println("Waiting as queue is empty..");
      try {
        list.wait();
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    }
    // if there is element in the list then retrieve it
    System.out.println("Consuming from queue- " +
Thread.currentThread().getName() + " " + list.remove());
    list.notify();
  }
 }
}
}
```

**RESULT**

The above program is executed and obtained the output.

**OUTPUT**



# PROGRAM 8

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a stack .

STEP 3: Push(); enter an element n

If top<n

Top++

Stack[top]=n.

STEP 4:Pop()

If top!=1

.top--

STEP 5: Print the stack.

STEP 6: Stop

**SOURCE CODE**

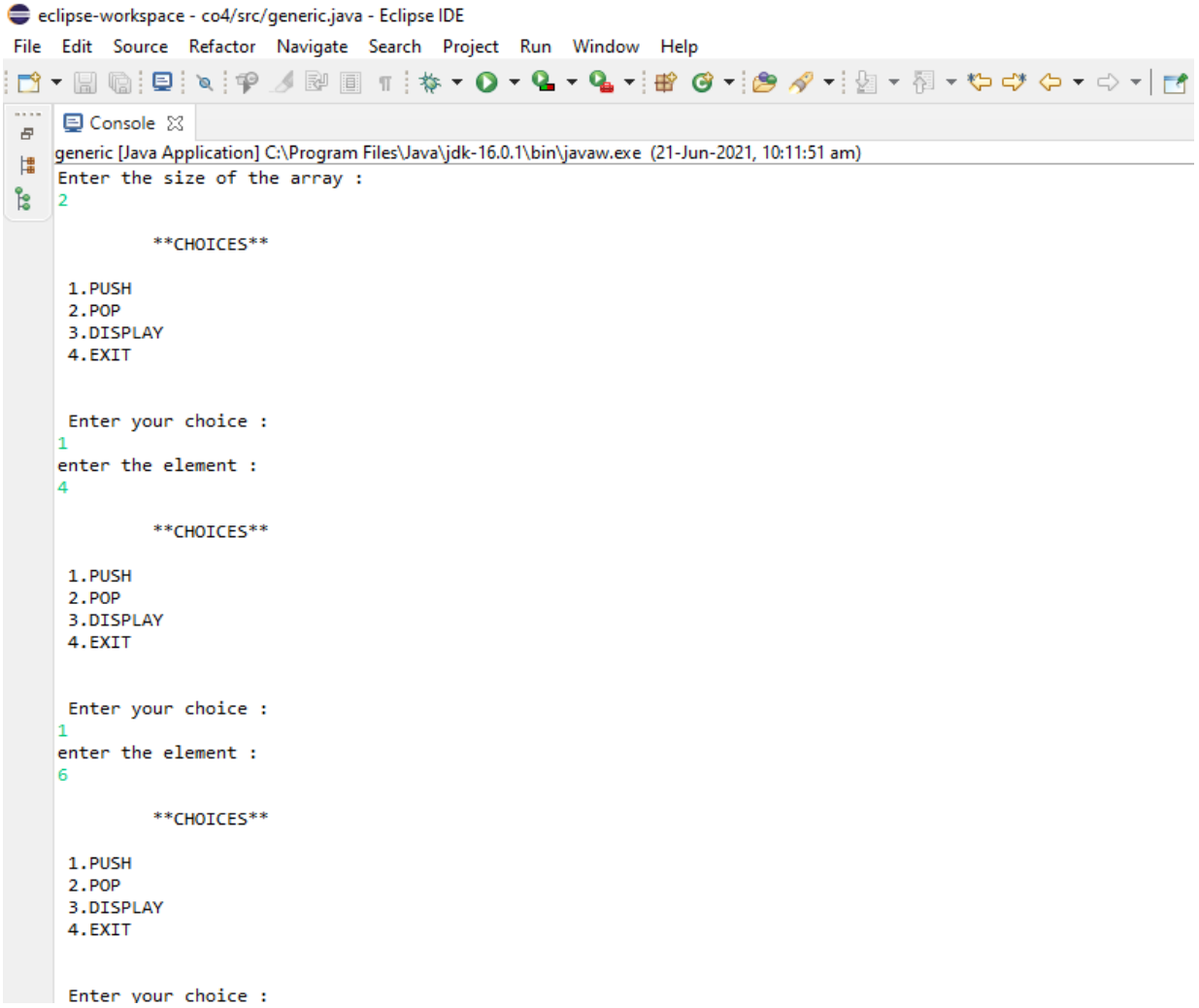
| | |
|---|---|
| generic.java | import java.util.Scanner;<br>class StackArr<br>{<br>int a[] = new int[20];<br>int top=-1,ch,item,i;<br>Scanner sc = new Scanner(System.in);<br>public void stackoperation()<br>{<br>System.out.println("Enter the size of the array : ");<br>int n=sc.nextInt();<br>do<br>{<br>System.out.println("\n\t **CHOICES**  ");<br>System.out.println("\n 1.PUSH \n 2.POP \n 3.DISPLAY \n 4.EXIT \n");<br>System.out.println("\n Enter your choice : ");<br>ch=sc.nextInt();<br>switch(ch)<br>{<br>case 1: if(top >=n-1)<br>{<br>System.out.println("stack overflow"); |

```
        }
        else
        {
        System.out.println("enter the element :");
        item =sc.nextInt();
        top=top+1;
        a[top]=item;
        }
        break;
case 2 : if(top<0)
         {
                        System.out.println("stack underflow");
         }
         else
         {
         a[top]='\0';
         top=top-1;
         }
         break;
case 3 :   if(top < 0)
                        {
                        System.out.println("\n stack is empty");
                        }
                        else
                        {
                        System.out.println("\n stack is \n");
                        for(i=top;i>=0;i--)
                        {
                        System.out.println(a[i]);
                        }
                        }
         break;
case 4 :
               System.out.println("\nTerminate the stack");
                break;
default : System.out.println("\n Invalid choice");
}
}
while(ch!=4);
}
}
class generic
{
public static void main(String[] args)
{
            StackArr sa =new StackArr();
            sa.stackoperation();
}}
```

## RESULT

The above program  is executed and obtained the output.

## OUTPUT



# PROGRAM 9

**AIM:** Using generic method perform Bubble sort.

**ALGORITHM**

STEP 1 : Start

STEP 2: Look at the first number in the list.

STEP 3: Compare the current number with the next number.

STEP 4: Is the next number smaller than the current number? If so, swap the two numbers around. If not, do not swap.

STEP 5: Move to the next number along in the list and make this the current number.

STEP 6: Repeat from step 2 until the last number in the list has been reached.

STEP 7: If any numbers were swapped, repeat again from step 1.

STEP 8: If the end of the list is reached without any swaps being made, then the list is ordered and the algorithm can stop.

STEP 9: Stop

**SOURCE CODE**

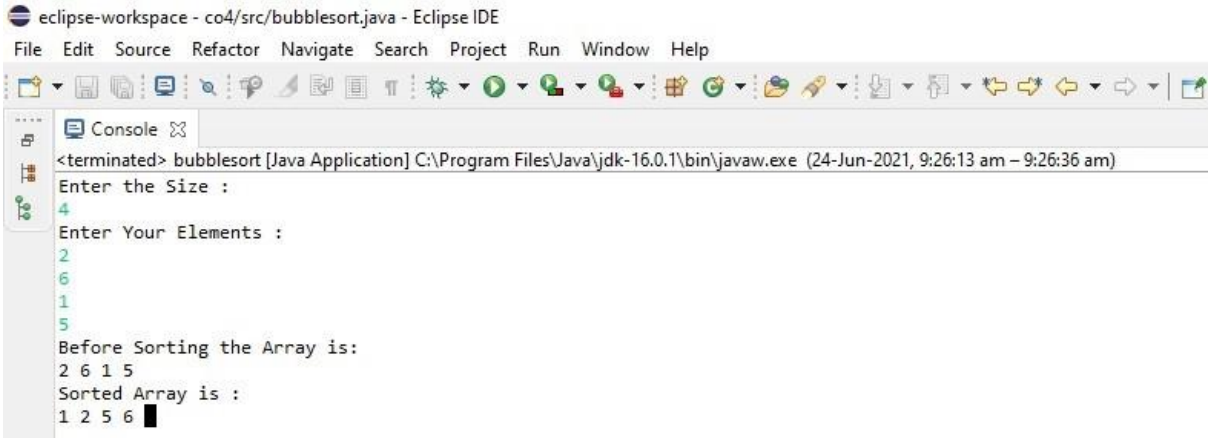
| | |
|---|---|
| bubblesort.java | import java.util.Scanner;<br>class bubble {<br><br>void sort(int ar[],int n) {<br>int temp;<br>int i,j;<br>for( i=0;i<ar.length-1;i++) {<br>for(j=0;j<ar.length-1-i;j++) {<br><br>if(ar[j]>ar[j+1]) {<br>temp=ar[j];<br>ar[j]=ar[j+1];<br>ar[j+1]=temp;<br>}}}}<br>void display(int ar[],int n) {<br>for (int i=0;i<n;i++) {<br>System.out.print(ar[i]+" ");<br>}}}<br>public class bubblesort {<br>public static void main(String[] args) { |

```
        int n,i;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the Size :");
        n=sc.nextInt();
        System.out.println("Enter Your Elements :");

        int ar[] = new int[n];
        for(i=0;i<n;i++)
        {
                ar[i] = sc.nextInt();
        }
        bubble  obj= new bubble  ();
        System.out.println("Before Sorting the Array is:");
        obj.display(ar,n);
         obj.sort(ar,n);

        System.out.println("\nSorted Array is :");
         obj.display(ar,n);
    }
}
```

**RESULT**

The above program  is executed and obtained the output.

**OUTPUT**



# PROGRAM 10

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Define an array .

STEP 3: Define objects to array.

STEP 4: Perform operation on array.

STEP 5: Print the array after operations.

STEP 6: Stop

**SOURCE CODE**

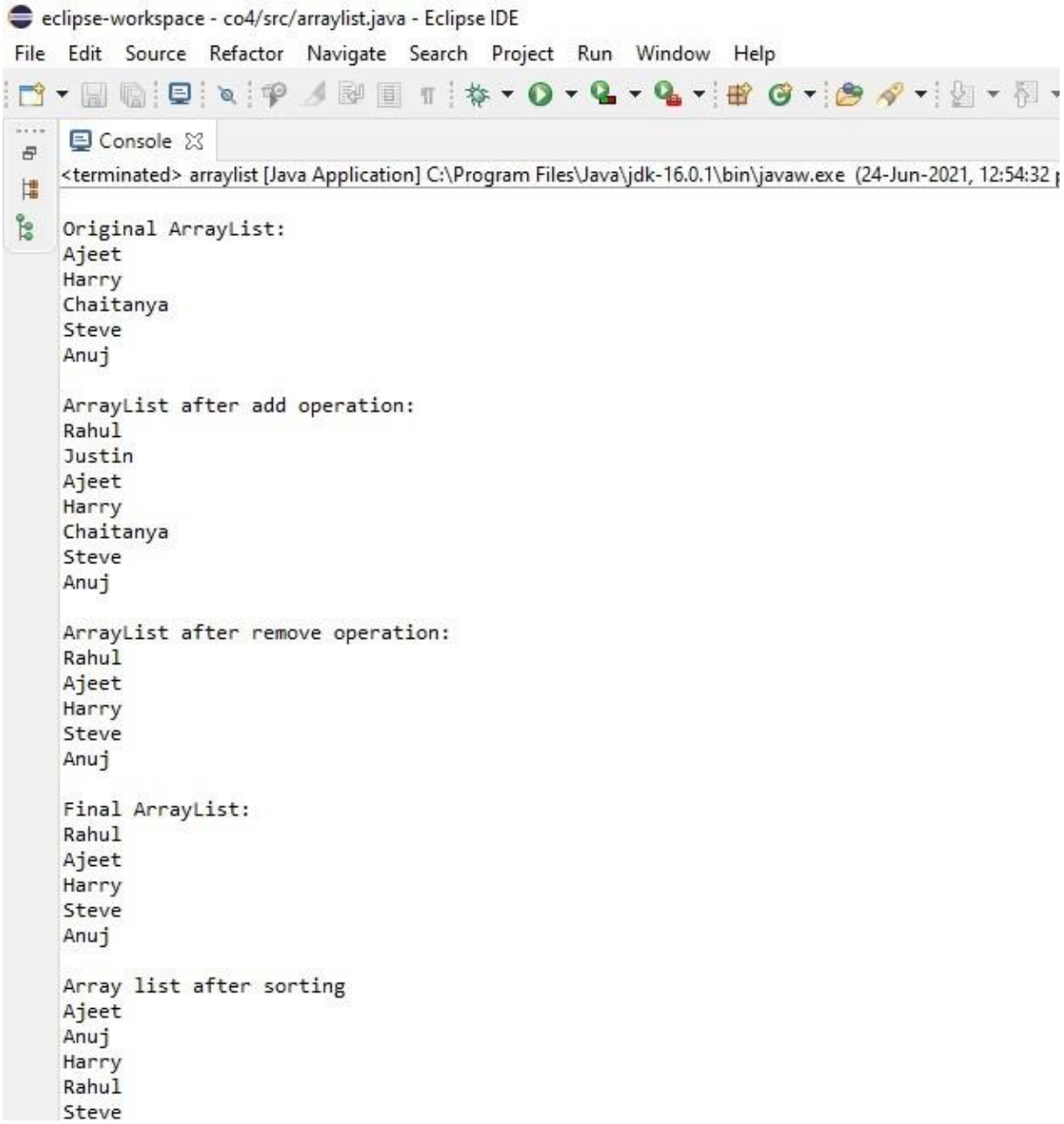
| | |
|---|---|
| arraylist.java | import java.util.*;<br><br>public class arraylist{<br>public static void main(String args[]) {<br><br>ArrayList<String> obj = new ArrayList<String>();<br>obj.add("Ajeet");<br>obj.add("Harry");<br>obj.add("Chaitanya");<br>obj.add("Steve");<br>obj.add("Anuj");<br>// Displaying elements<br>System.out.println("\nOriginal ArrayList:");<br>for(String str:obj)<br>System.out.println(str);<br>/* Add element at the given index*/<br>obj.add(0, "Rahul");<br>obj.add(1, "Justin");<br>// Displaying elements<br>System.out.println("\nArrayList after add operation:");<br>for(String str:obj)<br>System.out.println(str);<br>//Remove elements from ArrayList like this<br>obj.remove("Chaitanya");<br>//Remove element from the specified index<br>obj.remove(1);<br>// Displaying elements<br>System.out.println("\nArrayList after remove operation:");<br>for(String str:obj)<br>System.out.println(str);<br>// Displaying elements |

```
      System.out.println("\nFinal ArrayList:");
      for(String str:obj)
        System.out.println(str);
       //Sort the items//
      Collections.sort(obj);
      //Displaying elements//
      System.out.println(" \nArray list after sorting:");
      for(String str : obj)
      System.out.println(str);
       //Access the item//
      System.out.println("\nObject at index 2:"+obj.get(2));
       //Contain or not//
      System.out.println("\nRahul is in the
arraylist:"+obj.contains("Rahul"));
      System.out.println("\nJustin  is in the
arraylist:"+obj.contains("Justin"));
      //Size of list//
      System.out.println("\nSize of the arraylist :"+obj.size());
      // Clear all items//
      obj.clear();
      // Displaying elements
      System.out.println("\nArrayList after clear:"+obj);

    }
}
```

**RESULT**

The above program  is executed and obtained the output.

**OUTPUT**


eclipse-workspace - co4/src/arraylist.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console
<terminated> arraylist [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (24-Jun-2021, 12:54:32
Original ArrayList:
Ajeet
Harry
Chaitanya
Steve
Anuj
ArrayList after add operation:
Rahul
Justin
Ajeet
Harry
Chaitanya
Steve
Anuj
ArrayList after remove operation:
Rahul
Ajeet
Harry
Steve
Anuj
Final ArrayList:
Rahul
Ajeet
Harry
Steve
Anuj
Array list after sorting
Ajeet
Anuj
Harry
Rahul
Steve

eclipse-workspace - co4/src/arraylist.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Console

<terminated> arraylist [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (24-Jun-2021, 12:54:32 p

```
Justin
Ajeet
Harry
Chaitanya
Steve
Anuj

ArrayList after remove operation:
Rahul
Ajeet
Harry
Steve
Anuj

Final ArrayList:
Rahul
Ajeet
Harry
Steve
Anuj

Array list after sorting
Ajeet
Anuj
Harry
Rahul
Steve

Object at index 2:Harry

Rahul is in the arraylist:true

Justin  is in the arraylist:false

Size of the arraylist :5

ArrayList after clear:[]
```

# PROGRAM 11

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a linked list.

STEP 3: Define objects to Linked list using add().

STEP 4: Delete the elements in list using clear()

STEP 5: Print the Outputs.

STEP 6: Stop

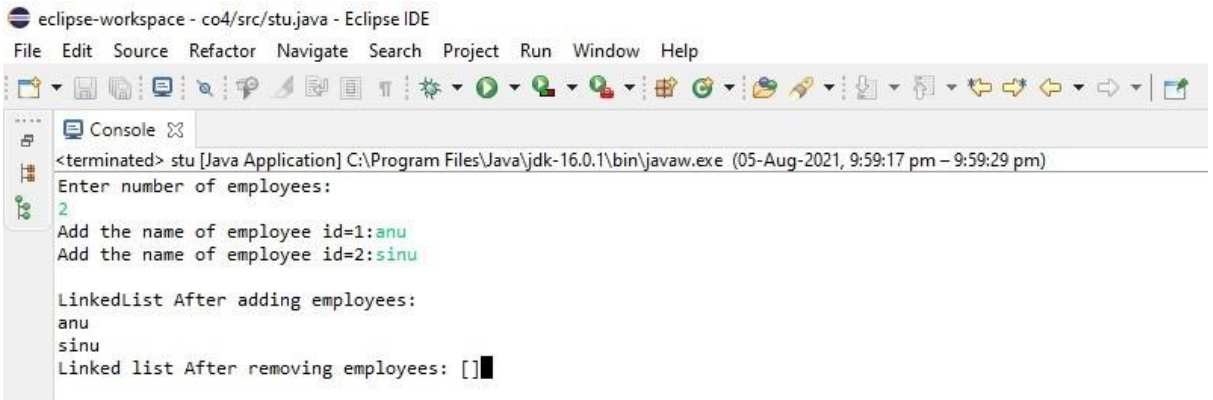
**SOURCE CODE**

| | |
|---|---|
| linked list.java | import java.util.*;<br>public class linked list {<br>public static void main(String[] args) {<br>LinkedList<String> list=new LinkedList<String>();<br>Scanner sc=new Scanner(System.in);<br>System.out.println("Enter number of employees:");<br>int num=sc.nextInt();<br>for(int i=1;i<=num;i++){<br>System.out.print("Add the name of employee id="+i+":");<br>String s=sc.next();<br>list.add(s);<br>}<br>System.out.println();<br>System.out.println("LinkedList After adding employees:");<br>Iterator<String> itr=list.iterator();<br>while(itr.hasNext()){<br>System.out.println(itr.next());<br>}<br>list.clear();<br>System.out.println("Linked list After removing employees: "+list);<br>}<br>} |

**RESULT**

The above program  is executed and obtained the output.

**OUTPUT**



## PROGRAM 12

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a Stack.

STEP 3: Define objects to  Stack using add().

STEP 4:Enter key to remove the item.

STEP 5: Remove the item using remove(key).

STEP 6: Print outputs

STEP 7: Stop

**SOURCE CODE**

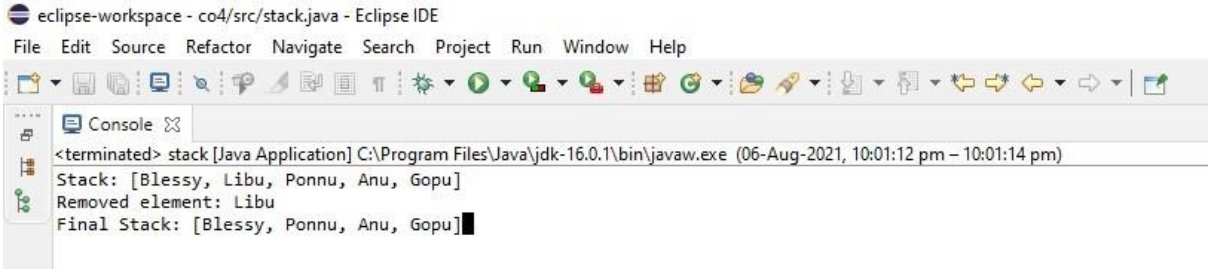
| | |
|---|---|
| stack.java | |

```
import java.util.*;
public class stack {
	public static void main(String args[])
	{
		// Creating an empty Stack
		Stack<String> stack = new Stack<String>();

		// Use add() method to add elements in the Stack
		stack.add("Blessy");
		stack.add("Libu");
		stack.add("Ponnu");
		stack.add("Anu");
		stack.add("Gopu");

		// Output the Stack
		System.out.println("Stack: " + stack);

		// Remove the element using remove()
		String rem = stack.remove(1);

		// Print the removed element
		System.out.println("Removed element: "+ rem);

		// Print the final Stack
		System.out.println("Final Stack: "+ stack);
	}
}
```

**RESULT**

The above program  is executed and obtained the output.

**OUTPUT**



# PROGRAM 13

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a Priority Queue.

STEP 3: Enter a limit n.

STEP 4: Add n elements to Queue.

STEP 5: Perform operations like remove(), add() etc.

STEP 6: Print outputs

STEP 7: Stop

**SOURCE CODE**

priQueue.java

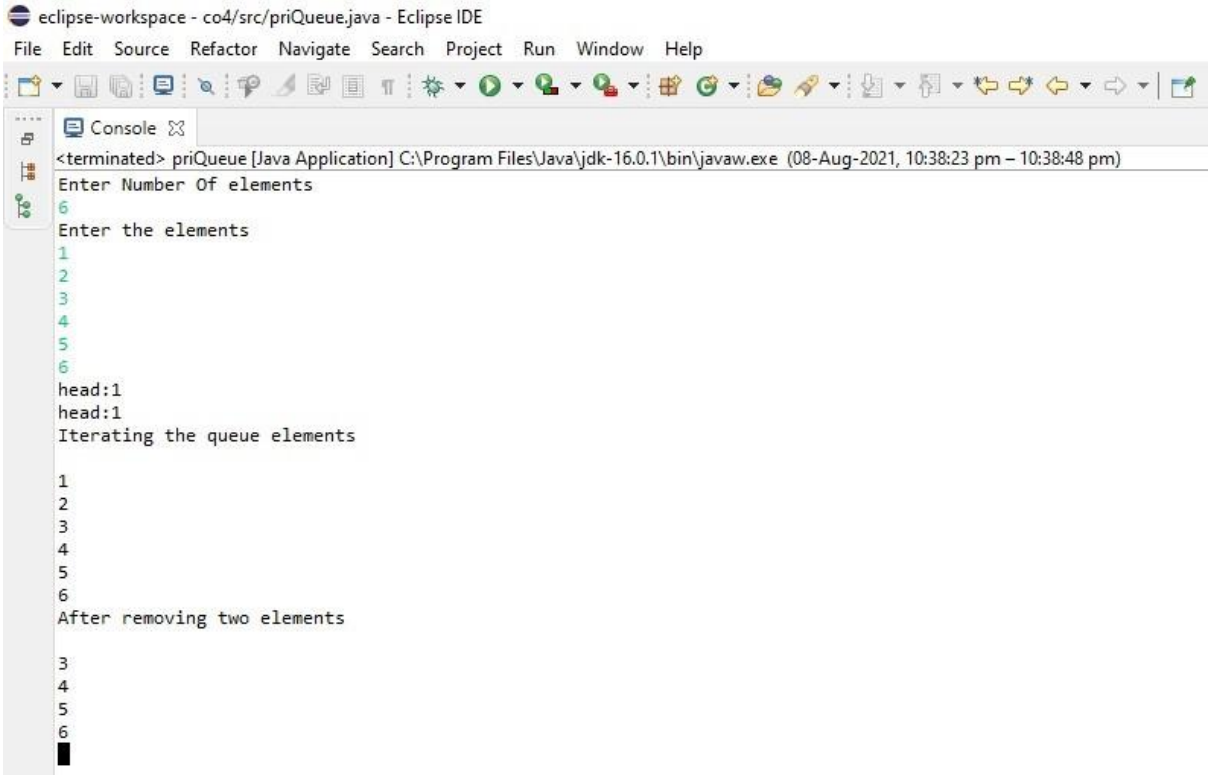
```
import java.util.Iterator;
import java.util.PriorityQueue;
import java.util.Scanner;
public class priQueue
{
	public static void main(String args[])
	{
		PriorityQueue<String> queue=new
PriorityQueue<String>();
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter Number Of elements
");
		int n=sc.nextInt();
		System.out.println("Enter the elements ");
		for(int i =0;i<n;i++)
		{
			String st=sc.next();
			queue.add(st);

		}
		System.out.println("head:"+queue.element());
		System.out.println("head:"+queue.peek());
		System.out.println("Iterating the queue
elements\n ");
		Iterator itr=queue.iterator();
		while(itr.hasNext()){
			System.out.println(itr.next());
		}
		queue.remove();
```

```
                queue.poll();
                System.out.println("After removing two
elements \n");
                Iterator<String> itr2=queue.iterator();
                while(itr2.hasNext()){
                        System.out.println(itr2.next());
                }
        }
}
```

**RESULT**

The above program  is executed and obtained the output.

**OUTPUT**



# PROGRAM 14

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a dequeue.

STEP 3: Insert the limit and numbers.

STEP 4:Perform pop() and remove() operations.

STEP 5: Print outputs

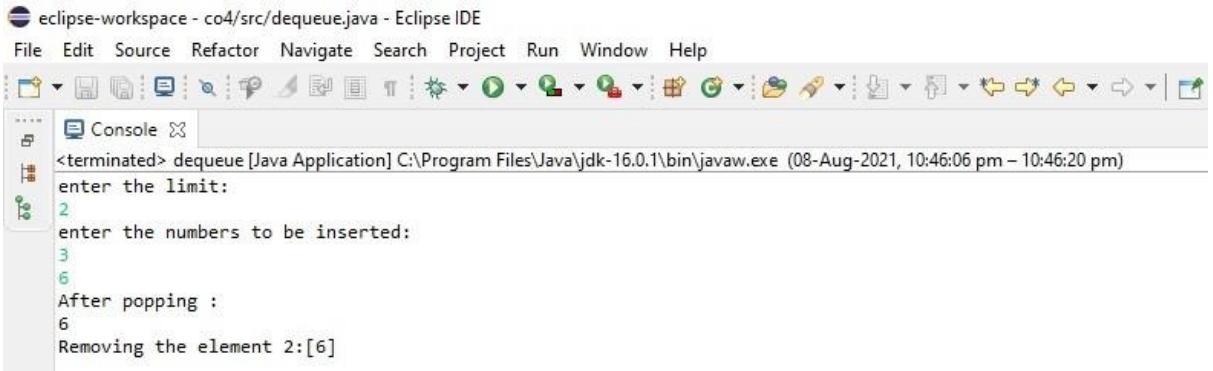
STEP 6: Stop

**SOURCE CODE:**

| | |
|---|---|
| dequeue.java | import java.util.*;<br>public class dequeue<br>{<br>public static void main(String[] args) {<br>Deque<Integer> deque = new ArrayDeque<Integer>();<br>// Inserts the elements<br>Scanner sc=new Scanner(System.in);<br>System.out.println("enter the limit:");<br>int n=sc.nextInt();<br>System.out.println("enter the numbers to be inserted: ");<br>for(int i =0;i<n;i++)<br>{<br>Integer obj=sc.nextInt();<br>deque.add(obj);<br><br>}<br><br>// Popping the element<br>deque.pop();<br>System.out.println("After popping : ");<br>for (Integer integer : deque) {<br>System.out.println(integer);<br>}<br>deque.remove(2);<br>System.out.println("Removing the element 2:"+deque);<br>}<br>} |

**RESULT :** The above program is executed and obtained the output.

**OUTPUT:**



# PROGRAM 15

**AIM:** Program to demonstrate the creation of Set object using the LinkedHashset class

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a hashset.

STEP 3: Define objects to Linked hash set using add().

STEP 4:Perform Desired operations like adding already existing elements.

STEP 5: Remove the item using remove(key).

STEP 6: Print outputs

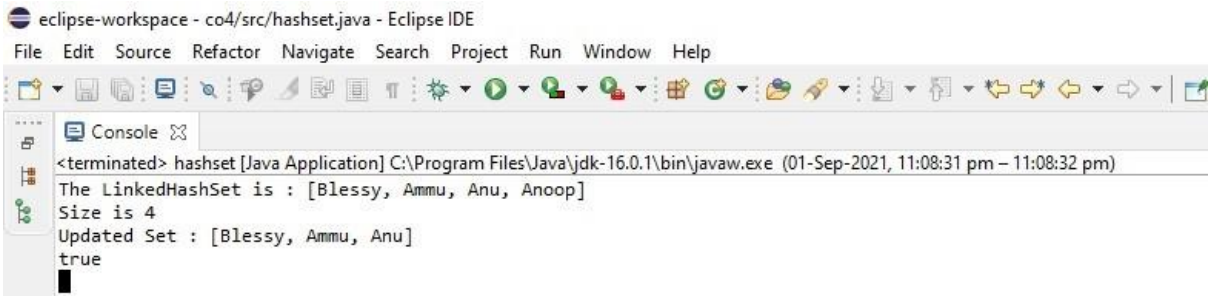
STEP 7: Stop

**SOURCE CODE:**

| | |
|---|---|
| hashset.java | import java.util.*;<br>public class hashset {<br>public static void main(String[] args) {<br><br>LinkedHashSet<String> l = new LinkedHashSet<String>();<br><br>// Inserting data<br>l.add("Blessy");<br>l.add("Ammu");<br>l.add("Anu");<br>l.add("Anoop");<br>System.out.println("The LinkedHashSet is : "+l);<br><br>//Size<br>System.out.println("Size is "+l.size());<br><br>//removing<br>l.remove("Anoop");<br>System.out.println("Updated Set : "+l);<br><br>//Checking<br>System.out.println(l.contains("Blessy"));<br><br>}<br>} |

**RESULT :** The above program is successfully executed and obtained the output.

**OUTPUT:**



# PROGRAM 16

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Define 2 hash sets h1 and h1.

STEP 3: Define objects to hash sets h_set and h_set2.

STEP 4: Compare the hash sets; if its same; .

STEP 5: Print the hashs are same

STEP 6: Else Print hashs are different

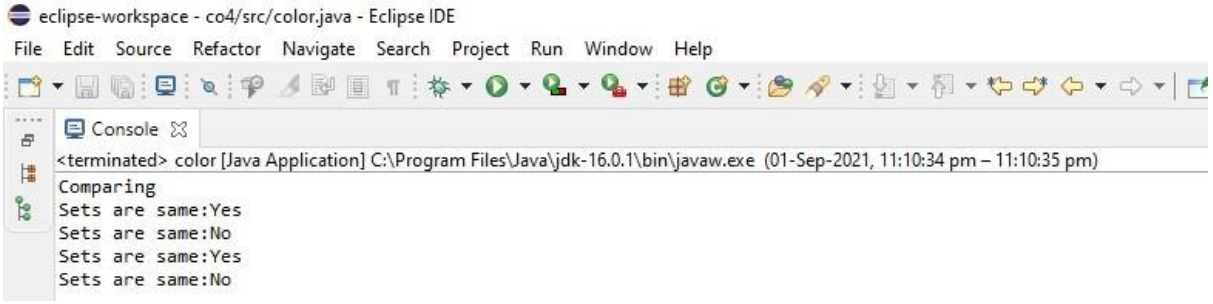
STEP 7: Stop

**SOURCE CODE:**

| | |
|---|---|
| color.java | import java.util.HashSet;<br><br>public class color {<br>public static void main(String[] args)<br>{<br><br>HashSet<String> h_set = new HashSet<String>();<br><br>h_set.add("Red");<br>h_set.add("Green");<br>h_set.add("Black");<br>h_set.add("White");<br><br>HashSet<String>h_set2 = new HashSet<String>();<br>h_set2.add("Red");<br>h_set2.add("Pink");<br>h_set2.add("Black");<br>h_set2.add("Orange");<br><br>System.out.println("Comparing");<br><br>HashSet<String>result_set = new HashSet<String>();<br>for (String element : h_set){<br>System.out.println(h_set2.contains(element) ? "Sets are same:Yes" : "Sets are same:No");<br>}<br>}<br><br>} |

**RESULT :** The above program is successfully executed and obtained the output.

**OUTPUT:**



# PROGRAM 17

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

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a Map set maphash.

STEP 3: Define objects to Map set maphash using put().

STEP 4: Remove the elements using remove(); .

STEP 5: Print the Outputs.

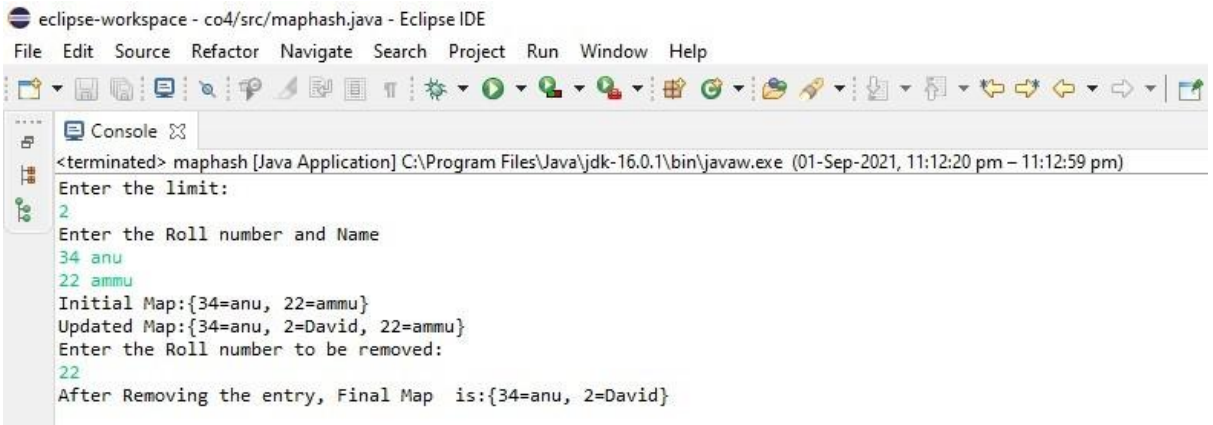
STEP 6: Stop

**SOURCE CODE:**

| | |
|---|---|
| maphash.java | import java.util.*;<br>public class maphash {<br>public static void main(String args[])<br>{<br>Map<Integer, String> mp = new HashMap<>();<br>//Inserting elements..<br>System.out.println("Enter the limit:");<br>Scanner inp = new Scanner(System.in);<br>int n= inp.nextInt();<br>System.out.println("Enter the Roll number and Name");<br>while(n!=0) {<br>int e= inp.nextInt();<br>String s= inp.next();<br>mp.put(e, s);<br>n--;<br>}<br>System.out.println("Initial Map:"+mp);<br>mp.put( (2), "David");<br>//Updating..<br>System.out.println("Updated Map:"+mp);<br>//Removing..<br>System.out.println("Enter the Roll number to be removed:");<br>int r=inp.nextInt();<br>mp.remove(r);<br>// Final Map..<br>System.out.println("After Removing the entry, Final Map is:"+mp);<br><br>}<br>} |

**RESULT :** The above program is successfully executed and obtained the output.

**OUTPUT:**



# PROGRAM 18

**AIM** : Program to Convert HashMap to TreeMap.

**ALGORITHM**

STEP 1 : Start

STEP 2: Define a map mp.

STEP 3: Define objects to mp using put().

STEP 4: Define a tree set treeMap

STEP 4:Convert map set into tree using treeMap.putAll(map).

STEP 5: Print the Outputs.

STEP 6: Stop

**SOURCE CODE:**

| | |
|---|---|
| maptree.java | import java.util.*;<br><br>public class maptree {<br>public static void main(String args[]) {<br><br>Map<String, String> map = new HashMap<>();<br>System.out.println("Enter the limit:");<br>Scanner inp = new Scanner(System.in);<br>int n= inp.nextInt();<br>System.out.println("Enter the Roll number and Name");<br>while(n!=0) {<br><br>String e= inp.next();<br>String s= inp.next();<br>map.put(e, s);<br>n--;<br>}<br><br>System.out.println("HashMap:"+map);<br>Map<String, String> treeMap = new TreeMap<>();<br>treeMap.putAll(map);<br>System.out.println("TreeMap:"+treeMap);<br>}<br>} |

**RESULT :** The above program is successfully executed and obtained the output.

**OUTPUT:**