20MCA132 - OBJECT ORIENTED PROGRAMMING LAB

CO 3 & CO 4

Submitted by,

Abhilash John

20MCA201

S2 MCA

LAB CYCLE 3

PROGRAM: 10

AIM : Area of different shapes using overloaded functions

ALGORITHM :

Step 1: Start

Step 2: Define the main class

Step 3: Define methods with the same methodname that performs the area operation for each shape

Step 4: Display the areas of each shapes.

Step 5: Stop

PROGRAM CODE :

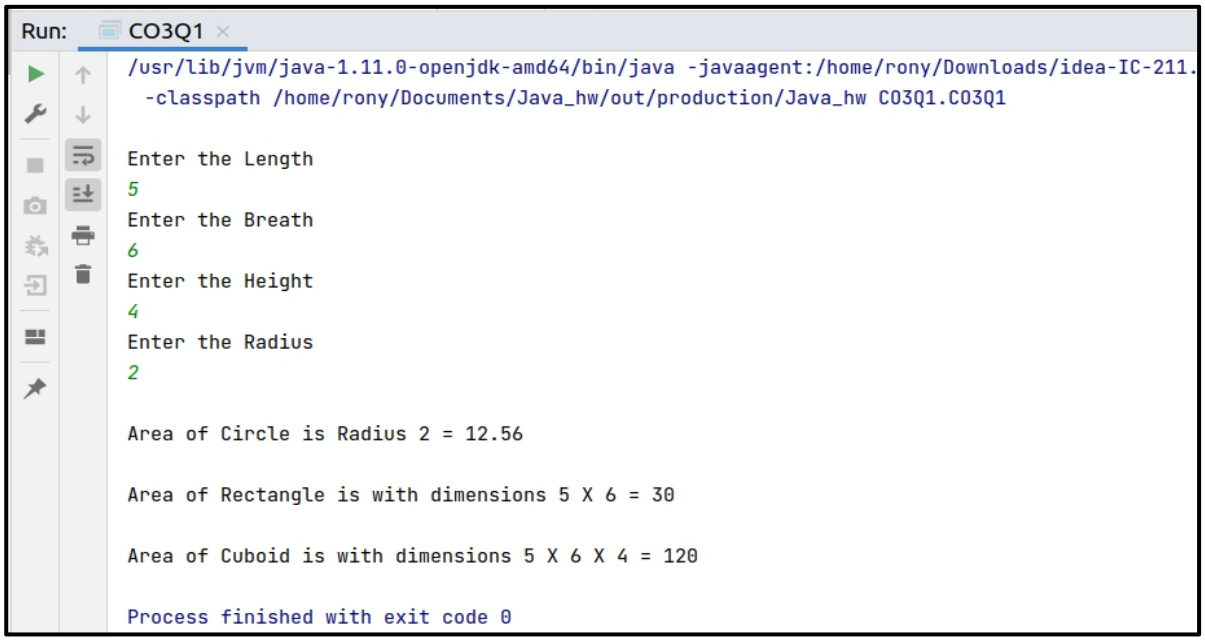
CO3Q1.java

```
package CO3Q1;
import java.util.Scanner;
public class CO3Q1 {
   void area(int r1){
      double Area_val = 3.14*r1*r1;
      System.out.println("\nArea of Circle is Radius "+r1+" = "+Area_val);
   }
   void area(int a1,int b1){
      int Area_val = a1*b1;
      System.out.println("\nArea of Rectangle is with dimensions "+a1+" X
"+b1+" = "+Area_val);
   }
   void area(int a1,int b1,int c1){
      int Area_val = a1*b1*c1;
      System.out.println("\nArea of Cuboid is with dimensions "+a1+" X
"+b1+" X "+c1+" = "+Area_val);
   }
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.println("\nEnter the Length");
      int l = sc.nextInt();
      System.out.println("Enter the Breath");
      int b = sc.nextInt();
      System.out.println("Enter the Height");       int h = sc.nextInt();
      System.out.println("Enter the Radius");
      int r = sc.nextInt();
```

```
        CO3Q1 obj1 = new CO3Q1();
        obj1.area(r);
        obj1.area(l,b);
        obj1.area(l,b,h);
    }
}
```

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

OUTPUT :



PROGRAM: 11

AIM : Create a class 'Employee' with data members Empid, Name, Salary, Address and constructors to initial ize 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.

ALGORITHM :

Step 1: Start

Step 2: create class "employee" with the provided data members and define the constructors

Step 3: create another class "Teachers" that performs inheritance of employee class and define constructors for the same

Step 4: create an array of objects in the corresponding class

Step 5: Display the details for the number of teachers provided

Step 6: Stop

PROGRAM CODE :

Teacher.java	package CO3Q2;
	
	import java.util.Scanner;
	class Employee {
	int Empid;
	String Name;
	double Salary;
	String Address;
	Employee(int no, String na, double sal, String add) {
	this.Empid = no;
	this.Name = na;
	this.Salary = sal;
	this.Address = add;
	}
	}
	public class Teacher extends Employee{
	String dept;
	String subject;
	
	Teacher(int no, String na, double sal, String add, String dep, String sub){

```
    super(no,na,sal,add);
    this.dept= dep;
    this.subject=sub;
  }

  void display(){
    System.out.println("Employee id: "+Empid);
    System.out.println("Name: "+Name);
    System.out.println("Salary: "+Salary);
    System.out.println("Address: "+Address);
    System.out.println("Department: "+dept);
    System.out.println("Subject: "+subject);
  }
  public static void main(String[] args) {
    System.out.println("\nEnter the No. of Employee's");
    Scanner sc1 = new Scanner(System.in);
    int num = sc1.nextInt();
    Teacher arr[]=new Teacher[num];
    for(int i =0;i<num;i++)
    {
       Scanner sc =new Scanner(System.in);
       System.out.println("Enter Employee id: ");
       int Empid=sc.nextInt();
       System.out.println("Enter Employee Name: ");
       String Name=sc.next();
       System.out.println("Enter Salary: ");
       double Salary=sc.nextDouble();
       System.out.println("Enter Address: ");
       String Address=sc.next();
       System.out.println("Enter department: ");
       String dept=sc.next();
       System.out.println("Enter Subject: ");
       String subject=sc.next();
       arr[i]=new Teacher(Empid,Name,Salary,Address,dept,subject);
       sc.close();
    }
    System.out.println("\n********Informations of all the
employee's************");
    for(int i=0;i<num;i++){
       int j=i+1;
       System.out.println("\n"+j+").");
       arr[i].display();
  }
  sc1.close();
  }
}
```

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

OUTPUT :


Run:
Teacher
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/rony/Downloads/idea-IC-211.7442.40/lib/idea_rt.jar=37705:/home/rony/Downloads/idea-IC-211.7442.40/bin -Dfile.encoding=UTF-8
-classpath /home/rony/Documents/Java_hw/out/production/Java_hw C03Q2.Teacher
Enter the No. of Employee's
1
Enter Employee id:
22
Enter Employee Name:
quion
Enter Salary:
45566778
Enter Address:
tre
Enter department:
mech
Enter Subject:
mech
********Informations of all the employee's**********
1).
Employee id: 22
Name: quion
Salary: 4.5566778E7
Address: tre
Department: mech
Subject: mech

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' that inherits 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 contains its 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.

ALGORITHM :

Step 1: Start

Step 2: create class "person" with the provided data members and define the constructors

Step 3: create another class "employee" that performs inheritance of person class and another class "teacher" that further inherits the properties of its former class

Step 4: create an array of objects in the corresponding class

Step 5: Display the details for the number of teachers provided

Step 5:Stop

PROGRAM CODE :

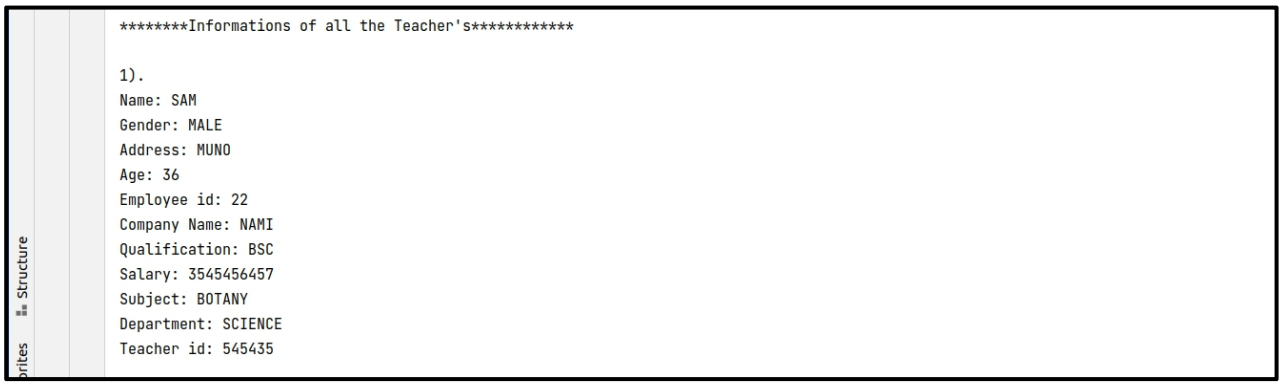
Teacher2.java	package CO3Q3;// Create a class 'Person' with data members Name, Gender, Address, Age and a constructor
	import java.util.Scanner;
	class person {
	String Name;
	String Gender;
	String Address;
	int Age;
	person(String name,String gender,String address, int age) {
	this.Name = name;
	this.Gender = gender;
	this.Address = address;
	this.Age = age;
	}
	}
	{
	int Empid;
	String Company_name;
	String Qualification;
	long Salary;

```
 Employee(String name,String gender,String address, int age,int empid,
String company_name, String qualification,long salary)
 {
    super(name,gender,address,age);
    this.Empid= empid;
    this.Company_name=company_name;
    this.Qualification=qualification;
    this.Salary=salary;
 }
}
 public class Teacher2 extends Employee{
    String Subject;
    String Department;
    String Teacherid;
   Teacher2(String name,String gender,String address, int age,int empid,
String company_name, String qualification,long salary, String subject, String
department, String teacherid){

super(name,gender,address,age,empid,company_name,qualification,salary);
       this.Subject=subject;
       this.Department=department;
       this.Teacherid=teacherid;
   }

   void display(){
       System.out.println("Name: "+Name);
       System.out.println("Gender: "+Gender);
       System.out.println("Address: "+Address);
       System.out.println("Age: "+Age);
       System.out.println("Employee id: "+Empid);
       System.out.println("Company Name: "+Company_name);
       System.out.println("Qualification: "+Qualification);
       System.out.println("Salary: "+Salary);
       System.out.println("Subject: "+Subject);
       System.out.println("Department: "+Department);
       System.out.println("Teacher id: "+Teacherid);

    }
    public static void main(String[] args) {
       System.out.println("Enter the No. of Teacher's");
       Scanner sc1 = new Scanner(System.in);
       int num = sc1.nextInt();
       Teacher2 arr[]=new Teacher2[num];
       System.out.println("Enter the Teacher Details");
       int x = 0,j=0;
       Scanner sc =new Scanner(System.in);
       for(int i =0;i<num;i++)
       {
          x = i +1;
          System.out.println(""+x+").");
```

```
            System.out.println(" Name: ");
            String a =sc.next();
            System.out.println("Gender: ");
            String b =sc.next();
            System.out.println("Address: ");
            String c =sc.next();
            System.out.println("Age: ");
            int d =sc.nextInt();
            System.out.println("Employee id: ");
            int e =sc.nextInt();
            System.out.println("Company name: ");
            String f =sc.next();
            System.out.println("Qualification: ");
            String g =sc.next();
            System.out.println("Salary: ");
            long h =sc.nextLong();
            System.out.println(" Subject: ");
            String k =sc.next();
            System.out.println("Department: ");
            String l =sc.next();
            System.out.println("Teacher Id: ");
            String n =sc.next();
            arr[i]=new Teacher2(a,b,c,d,e,f,g,h,k,l,n);
        }
        sc.close();
        System.out.println("\n********Informations of all the
Teacher's***********");
        for(int i=0;i<num;i++){
            j=i+1;
            System.out.println("\n"+j+").");
            arr[i].display();
    }
    sc1.close();
    }
}
```

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

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.

ALGORITHM :

Step 1: Start

Step 2: create class “publisher” and initialize its data members

Step 3: create classes book, literature, fiction. Each class inherit from their subsequent previous class and have its own data members

Step 4: create an array of objects in the corresponding class

Step 5: Display the details of the books require

Step 6: Stop

PROGRAM CODE :

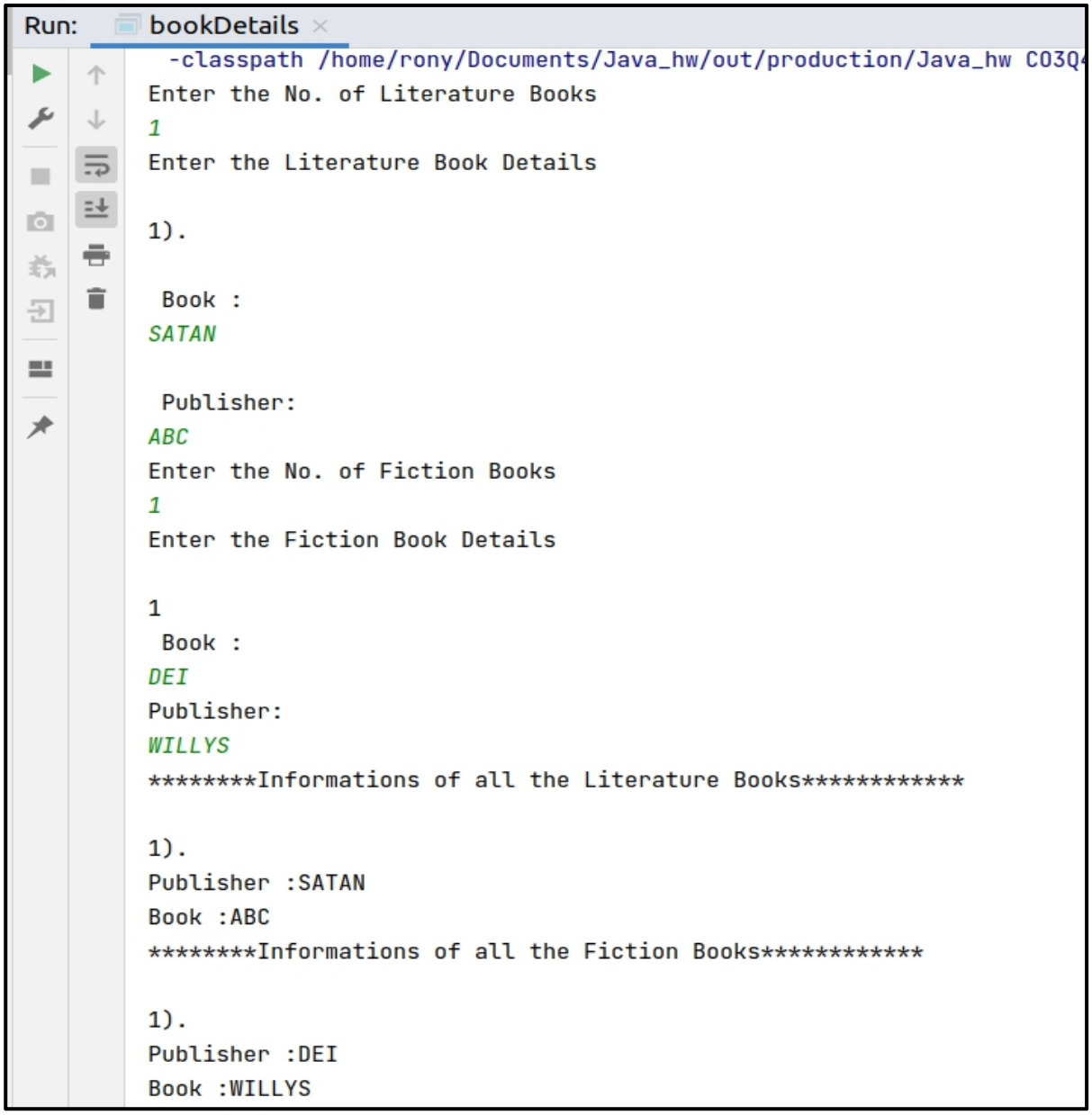
bookDetails.java	package CO3Q4;
	import java.util.Scanner;
	class Publisher{
	String publisher;
	Publisher(String pub){
	this.publisher=pub;
	}}
	class Book extends Publisher{
	String book;
	Book(String pub,String boo){
	super(pub);
	book=boo;
	}}
	class Literature extends Book{
	String category;
	Literature(String pub, String boo){
	super(pub, boo);
	}
	void display(){
	System.out.println("Publisher :"+publisher);
	System.out.println("Book :"+book);
	}}
	class Fiction extends Book{
	Fiction(String pub, String boo){
	super(pub, boo);

```
    }
    void display(){
        System.out.println("Publisher :"+publisher);
        System.out.println("Book :"+book);
    }}
public class bookDetails{
    public static void main(String[] args) {
        System.out.println("Enter the No. of Literature Books");
        Scanner sc1 = new Scanner(System.in);
        int num = sc1.nextInt();
        Literature arr[]=new Literature[num];
        System.out.println("Enter the Literature Book Details");
        int x = 0,j=0;
        Scanner sc =new Scanner(System.in);
        for(int i =0;i<num;i++)
        {
            x = i +1;
            System.out.println("\n"+x+").");
            System.out.println("\n Book : ");
            String boo =sc.next();
            System.out.println("\n Publisher: ");
            String pub =sc.next();

            arr[i]=new Literature(boo,pub);
        }
        System.out.println("Enter the No. of Fiction Books");
        int num1 = sc1.nextInt();
        Fiction arr1[]=new Fiction[num1];
        System.out.println("Enter the Fiction Book Details\n");
        int x1 = 0,j1=0;
        for(int i =0;i<num1;i++)
        {
            x1 = i +1;
            System.out.println(x1);
            System.out.println(" Book : ");
            String boo =sc.next();
            System.out.println("Publisher: ");
            String pub =sc.next();

            arr1[i]=new Fiction(boo,pub);
        }
        sc.close();
        sc1.close();

        System.out.println("********Informations of all the Literature
Books**********");
        for(int i=0;i<num;i++){
            j=i+1;
            System.out.println("\n"+j+").");
            arr[i].display(); }
```

```
        System.out.println("********Informations of all the Fiction
Books***********");
        for(int i=0;i<num1;i++){
            j1=i+1;
            System.out.println("\n"+j1+").");
            arr1[i].display(); }
    sc1.close(); } }
```

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

OUTPUT :



PROGRAM: 14

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: Start

Step 2: Create can interface results.

Step 3: Create classes Student and Sports that implements the interface results

Step 4: Display the academic and sports score of the student

Step 5: Stop

PROGRAM CODE :

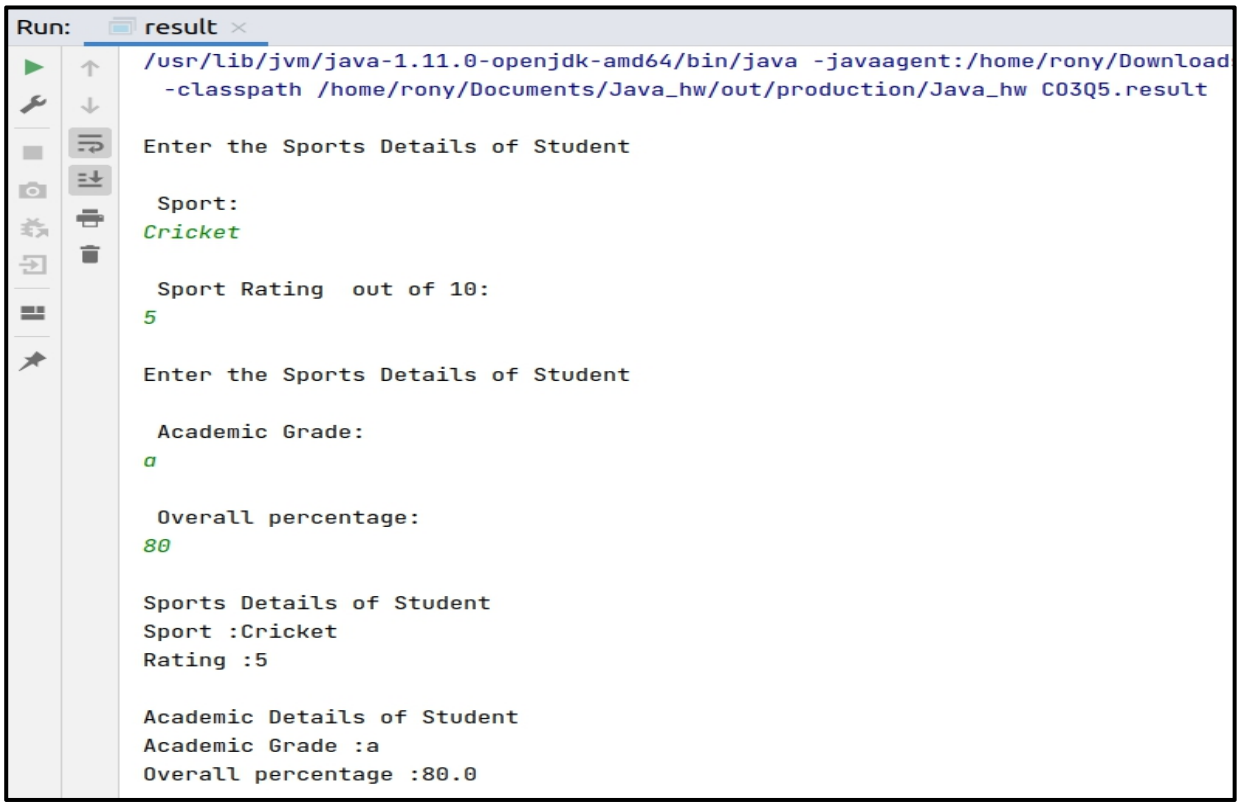
result.java

```
package CO3Q5;
import java.util.Scanner;
class sports{
   String sport;
   int Rating;
   sports(String spo, int ra){
      sport = spo;
      Rating = ra;
   }}
class student extends sports{
   String Grade;
   double Overall_per;
   student(String spo, int ra,String gd, double per ){
      super(spo, ra);
      Grade = gd;
      Overall_per = per;
   }}
public class result extends student {
   result(String spo, int ra,String gd, double per ){
      super(spo, ra, gd, per);
   }
   void display(){
      System.out.println("\nSports Details of Student");
      System.out.println("Sport :"+sport);
      System.out.println("Rating :"+Rating);
      System.out.println("\nAcademic Details of Student");
      System.out.println("Academic Grade :"+Grade);
      System.out.println("Overall percentage :"+Overall_per);
```

```
    }
    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 successfully executed and obtained the output

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.

ALGORITHM :

Step 1: Start

Step 2: Create an interface Calculation that has the methods to take inputs and compute area and perimeter

Step 3: Create classes Circle and Rectangle that implements calculation

Step 4: Display the area and perimeter of circle or rectangle depending upon the choice the user selects.

Step 5: Stop

PROGRAM CODE :

CO3Q6.java

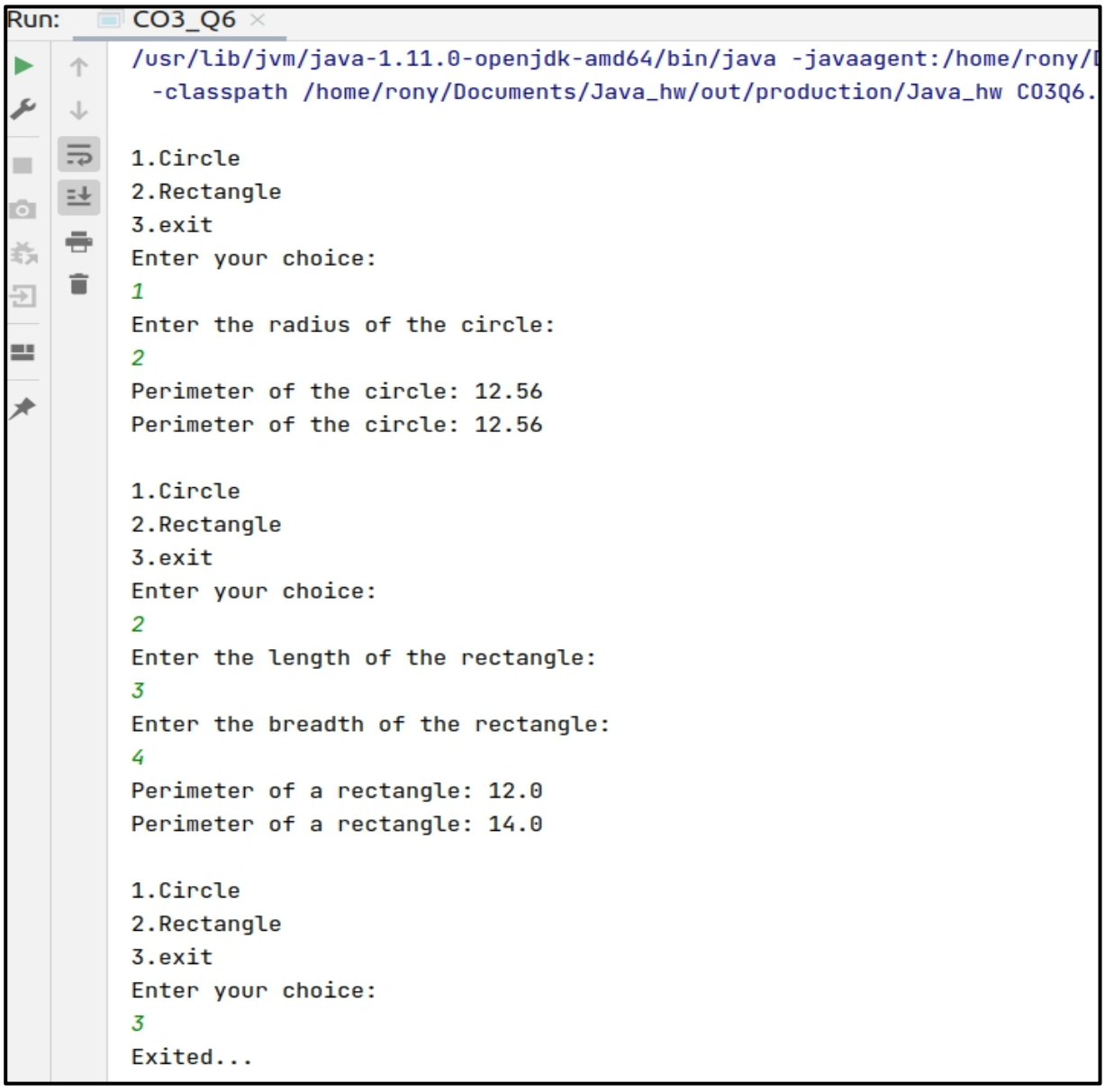
```
package CO3Q6;// Create an interface having prototypes of functions area() and perimeter(). Create two
// classes Circle and Rectangle which implements the above interface. Create a menu driven
// program to find area and perimeter of objects.
import java.util.Scanner;
interface prop
{
   void getdata();
   void area();
   void perimeter();
}
class Circle implements prop
{
   double pi = 3.14;
   double r;
   Scanner sc = new Scanner(System.in);
   @Override
   public void getdata()
   {
      System.out.println("Enter the radius of the circle:");
      r = sc.nextDouble();
   }
   @Override
   public void perimeter(){
      System.out.println("Perimeter of the circle: "+(2*pi*r));
```

```
    }
    @Override
    public void area()
    {
      System.out.println("Perimeter of the circle: "+(pi*r*r));
    }}
class Rectangle implements prop
{
    double l,b;
    Scanner sc = new Scanner(System.in);
    @Override
    public void getdata(){
      System.out.println("Enter the length of the rectangle:");
      l = sc.nextDouble();
      System.out.println("Enter the breadth of the rectangle:");
      b = sc.nextDouble();
    }
    @Override
    public void area()
    {
      System.out.println("Perimeter of a rectangle: "+(l*b));
    }
    @Override
    public void perimeter()
    {
      System.out.println("Perimeter of a rectangle: "+(2*(l+b)));
    }}
public class CO3Q6
{
    public static void main(String[] args)
    {
      int ch;
      Scanner sc = new Scanner(System.in);
      Circle ob = new Circle();
      Rectangle obj = new Rectangle();
      do
      {
        System.out.println("\n1.Circle\n2.Rectangle\n3.exit");
        System.out.println("Enter your choice:");
        ch = sc.nextInt();
        switch(ch)
        {
          case 1 :ob.getdata();
              ob.area();
              ob.perimeter();
              break;
          case 2 :obj.getdata();
              obj.area();
              obj.perimeter();
              break;
```

```
                case 3 :System.out.println("Exited...");
                    System.exit(0);
                }
            }while(true);
        }
    }
```

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

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

ALGORITHM :

Step 1: Start

Step 2: Create interface calc that perfroms the calculation operations.

Step 3: Create class bill that implements the interface calc

Step 4: Display the net amount by acquiring the data for the specific inputs

Step 5: Stop

PROGRAM CODE :

CO3Q7.java

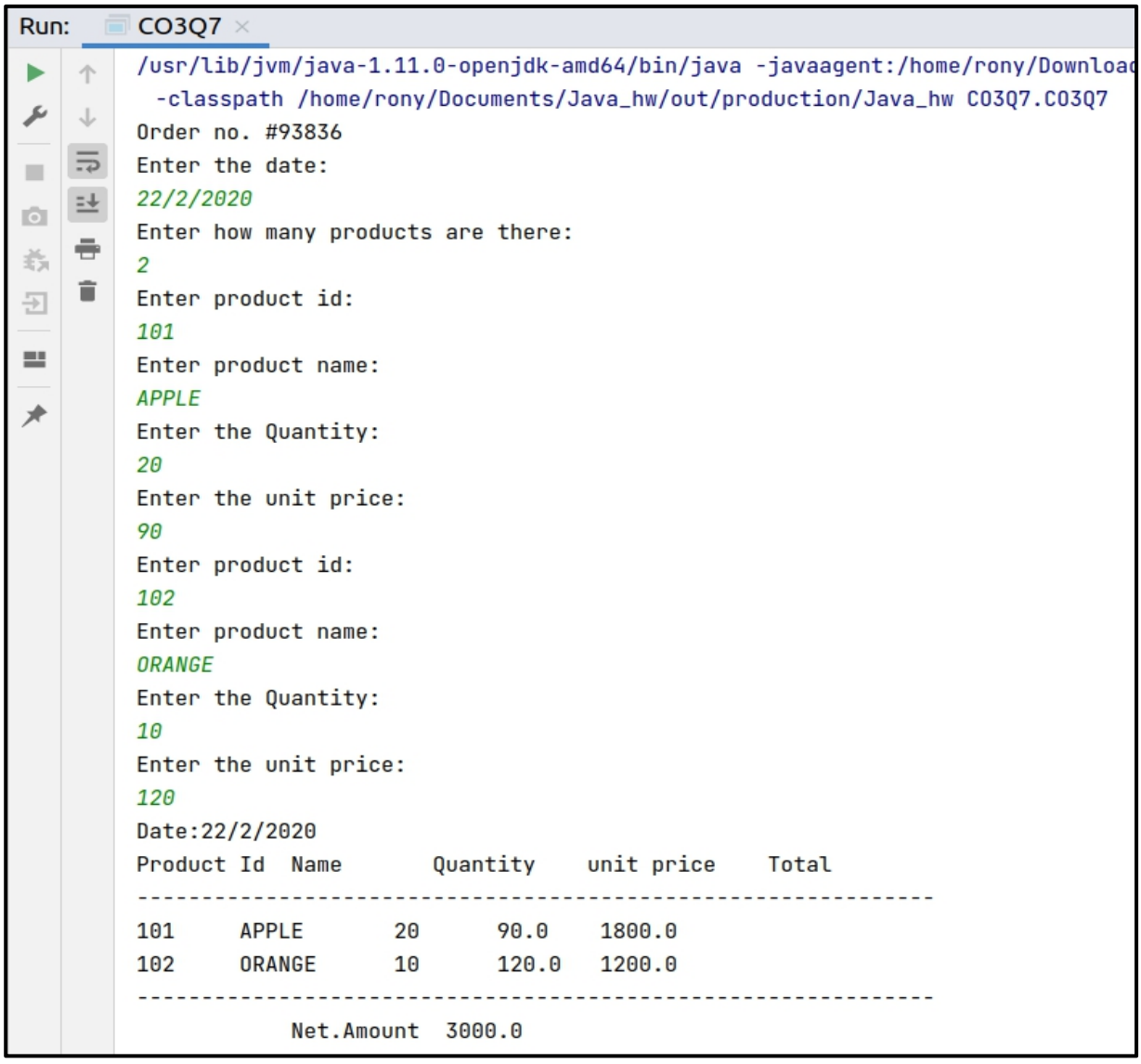
```
package CO3Q7;

import java.util.Scanner;

interface calc
{
   void calculate();
}

class bill implements calc
{
   String date,name,p_id;
   int quantity;
   double unit_price,total,namount=0;
   Scanner sc = new Scanner(System.in);
   public void getdata()
   {
      System.out.println("Enter product id:");
      p_id = sc.nextLine();
```

```
        System.out.println("Enter product name:");
        name = sc.nextLine();
        System.out.println("Enter the Quantity:");
        quantity = sc.nextInt();
        System.out.println("Enter the unit price:");
        unit_price = sc.nextDouble();
    }

    @Override
    public void calculate()
    {
        total = quantity * unit_price;
    }
    public void display()
    {

System.out.println(p_id+"\t\t"+name+"\t\t"+quantity+"\t\t"+unit_price+"\
t"+total);
    }
}


public class CO3Q7
{
    public static void main(String[] args)
    {
        int n,i;
        double namount=0,t;
        int ran;
        String date;
        t = Math.random() *1000000;
        ran = (int) t;
        Scanner sc = new Scanner(System.in);
        System.out.println("Order no. #"+ran);
        System.out.println("Enter the date:");
        date = sc.nextLine();
        System.out.println("Enter how many products are there:");
        n = sc.nextInt();
        bill ob[] = new bill[n];
        for(i=0;i<n;i++)
            ob[i] = new bill();
        for(i=0;i<n;i++){
            ob[i].getdata();
            ob[i].calculate();
        }
        System.out.println("Date:"+date);
        System.out.println("Product Id \tName\t   Quantity\t   unit price\t
Total ");
        System.out.println("---------------------------------------------------------");
        for(i=0;i<n;i++){
```

```
            ob[i].display();
            namount += ob[i].total;
        }
        System.out.println("-----------------------------------------------------------");
        System.out.println("\t\t\tNet.Amount\t"+ namount);

    }
}
```

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

OUTPUT :



LAB CYCLE 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..

ALGORITHM :

Step 1: Start

Step 2: To create a package named graphics, create a folder of the same name in the directory. Here inside that we have another module named calculate

Step 3: Inside the graphics folder, create modules for finding the areas of rectangle, circle, triangle and square.

Step 4: Outside the graphics folder, write a program to access the modules mention above and print the output

Step 5: Stop

PROGRAM CODE :

driver.java	//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.
	package CO4Q1;
	import CO4Q1.graphics.circle;
	import CO4Q1.graphics.rectangle;
	import CO4Q1.graphics.square;
	import CO4Q1.graphics.triangle;
	import java.util.Scanner;
	
	public class driver {
	  public static void main(String[] args) {
	  Scanner sc = new Scanner(System.in);
	  int choice;
	  circle obj1 = new circle();
	  rectangle obj2 = new rectangle();
	  square obj3 = new square();
	  triangle obj4 = new triangle();
	  System.out.println("Choose any 1)Circle 2)Rectangle 3)Square 4)Triangle: ");
	  choice = sc.nextInt();
	  switch (choice) {

	case 1:
	obj1.area();
	break;
	case 2:
	obj2.area();
	break;
	case 3:
	obj3.area();
	break;
	case 4:
	obj4.area();
	default:
	break;}} }

circle.java	package CO4Q1.graphics;
	import java.util.Scanner;
	public class circle implements area_cal{
	int radius;
	@Override
	public void area() {
	Scanner sc = new Scanner(System.in);
	System.out.println("Input radius of circle : ");
	radius = sc.nextInt();
	String area = Double.toString(Math.PI*radius*radius);
	System.out.println("Area of the circle is : "+area);
	sc.close();
	}}

rectangle.java	package CO4Q1.graphics;
	
	import java.util.Scanner;
	public class rectangle implements area_cal {
	int l,b;
	@Override
	public void area(){
	Scanner sc = new Scanner(System.in);
	System.out.println("Enter the length of the rectangle :");
	l = sc.nextInt();
	System.out.println("Enter the breath of the rectangle");
	b = sc.nextInt();
	System.out.println("Area of the rectangle = "+l*b);
	}
	}

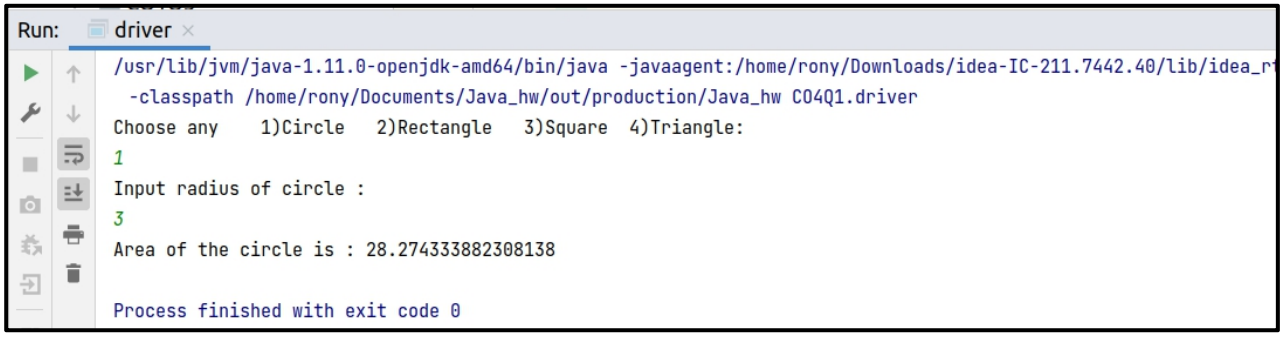
square.java	package CO4Q1.graphics;
	import java.util.Scanner;
	public class square implements area_cal{
	int side;
	@Override
	public void area() {
	Scanner sc = new Scanner(System.in);
	System.out.println("Input side length of square : ");
	side = sc.nextInt();
	String area = Double.toString(side*side);
	System.out.println("Area of the square : "+area);
	}
	}

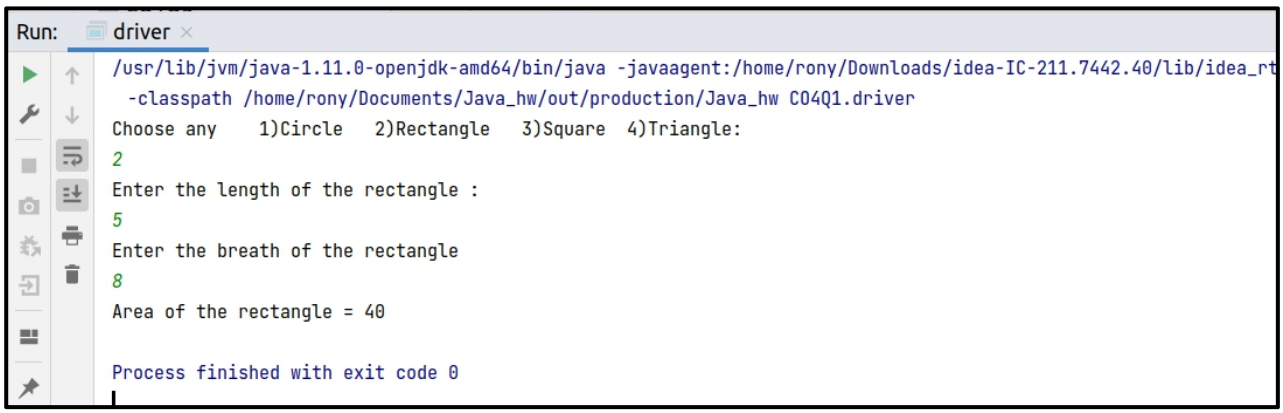
triangle.java	package CO4Q1.graphics;
	import java.util.Scanner;
	public class triangle implements area_cal{
	int height;
	int breadth;
	@Override
	public void area() {
	Scanner sc = new Scanner(System.in);
	System.out.println("Input height of the triangle : ");
	height = sc.nextInt();
	System.out.println("Input breadth of triangle : ");
	breadth = sc.nextInt();
	String area = Double.toString((height*breadth)/2f);
	System.out.println("Area of the triangle is : "+area);
	}
	}

area_cal.java	package CO4Q1.graphics;
	
	public interface area_cal {
	void area();
	}

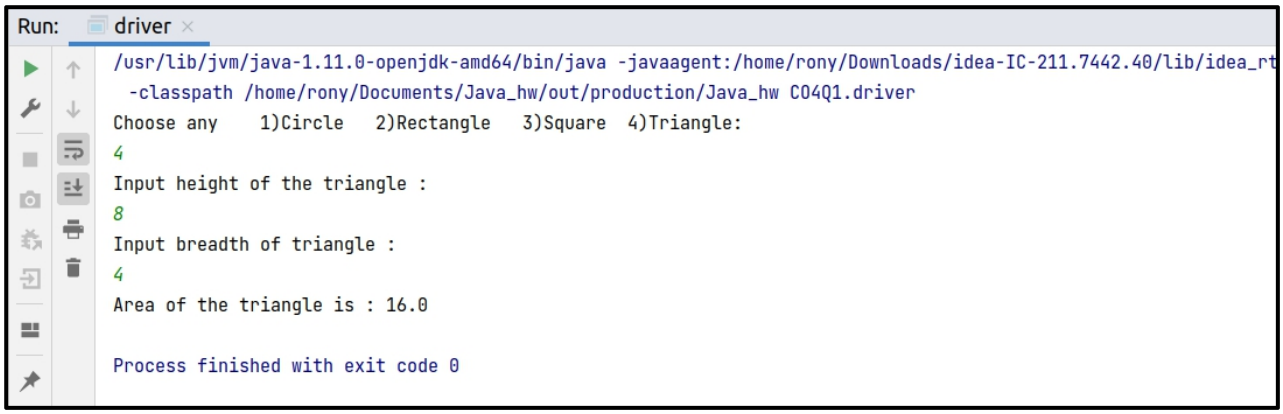
RESULT : The above program is successfully executed and obtained the output

OUTPUT :


Run: driver
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/rony/Downloads/idea-IC-211.7442.40/lib/idea_rt
-classpath /home/rony/Documents/Java_hw/out/production/Java_hw CO4Q1.driver
Choose any 1)Circle 2)Rectangle 3)Square 4)Triangle:
1
Input radius of circle :
3
Area of the circle is : 28.274333882308138
Process finished with exit code 0


Run: driver
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/rony/Downloads/idea-IC-211.7442.40/lib/idea_rt
-classpath /home/rony/Documents/Java_hw/out/production/Java_hw CO4Q1.driver
Choose any 1)Circle 2)Rectangle 3)Square 4)Triangle:
2
Enter the length of the rectangle :
5
Enter the breath of the rectangle
8
Area of the rectangle = 40
Process finished with exit code 0


Run: driver
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/rony/Downloads/idea-IC-211.7442.40/lib
-classpath /home/rony/Documents/Java_hw/out/production/Java_hw CO4Q1.driver
Choose any 1)Circle 2)Rectangle 3)Square 4)Triangle:
3
Input side length of square :
5
Area of the square : 25.0
Process finished with exit code 0


Run: driver
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/rony/Downloads/idea-IC-211.7442.40/lib/idea_rt
-classpath /home/rony/Documents/Java_hw/out/production/Java_hw CO4Q1.driver
Choose any 1)Circle 2)Rectangle 3)Square 4)Triangle:
4
Input height of the triangle :
8
Input breadth of triangle :
4
Area of the triangle is : 16.0
Process finished with exit code 0

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

ALGORITHM :

Step 1: Start

Step 2: To create a package named arithmetic, create a folder of the same name in the directory. Here inside that we have another module named operation

Step 3: Inside arithmetic package, create modules to perform addition, subtraction, multiplication and division of 2 numbers.

Step 4: Outside the folder, write another program that acssess the above module and print the output.

Step 5:Stop

PROGRAM CODE :

driver.java	package CO4Q2;
	import CO4Q2.operations.*;
	import java.util.Scanner;
	public class driver {
	public static void main(String[] args) {
	Scanner sc = new Scanner(System.in);
	System.out.println("Enter the First Number");
	int a = sc.nextInt();
	System.out.println("Enter the Second Number");
	int b = sc.nextInt();
	add obj1 =new add();
	obj1.cal(a,b);
	subtract obj2 =new subtract();
	obj2.cal(a,b);
	multiply obj3 = new multiply();
	obj3.cal(a,b);
	divide obj4 = new divide();
	obj4.cal(a,b);}}

add.java	package CO4Q2.operations;
	
	public class add implements calculate{
	public void cal(int x, int y){
	int sum = x+y;
	System.out.println("Sum of Numbers = "+sum);

calculate.java	package CO4Q2.operations;

	public interface calculate {
	  void cal(int x, int y);
	}

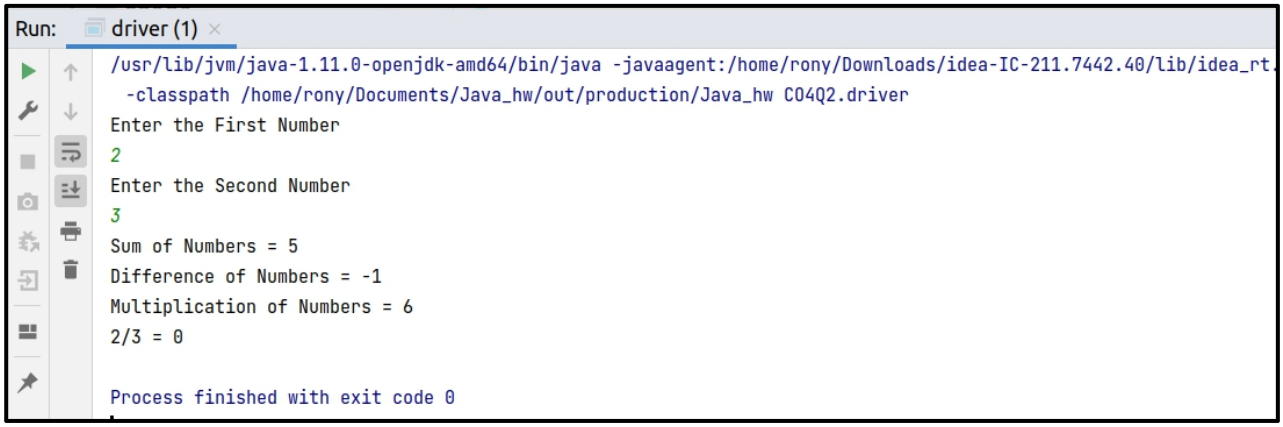
divide.java	package CO4Q2.operations;
	
	public class divide implements calculate{
	  public void cal(int x, int y){
	  int div = x/y;
	  System.out.println(x+"/"+y+" = "+div);
	}}

multiply.java	package CO4Q2.operations;
	
	public class multiply implements calculate{
	  public void cal(int x, int y){
	  int mul = x*y;
	  System.out.println("Multiplication of Numbers = "+mul);
	  }
	}

subtract.java	package CO4Q2.operations;
	public class subtract implements calculate{
	  public void cal(int x, int y){
	  int sub = x-y;
	  System.out.println("Difference of Numbers = "+sub);
	  }
	}

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

OUTPUT :



PROGRAM: 19

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

ALGORITHM :

Step 1: Start

Step 2: Create a class named usernameex that inherits Exception class with a constructor that calls Exception class constructor and pass error meaasage.

Step 3: Create a class named passwordex that inherits Exception class with a constructor that calls Exception class constructor and pass error meaasage.

Step 4: Inside the main(), Read the username and password.

Step 5: Inside the try block, we throw usernamex and passwordex with appropriate message if any of the conditon is true:

a. If username is empty

b. If password is empty

c. If password doesnt contain special charecters

d. If username length is less than 6

e. If password is not string enough

Step 6: Inside the catch block with parameter usernameex's object, print "USERNAME EXCEPTION OCCURED"

Step 7: Inside the catch block with parameter passwordex's object, print "PASSWORD EXCEPTION OCCURED"

Step 8:Stop

PROGRAM CODE :

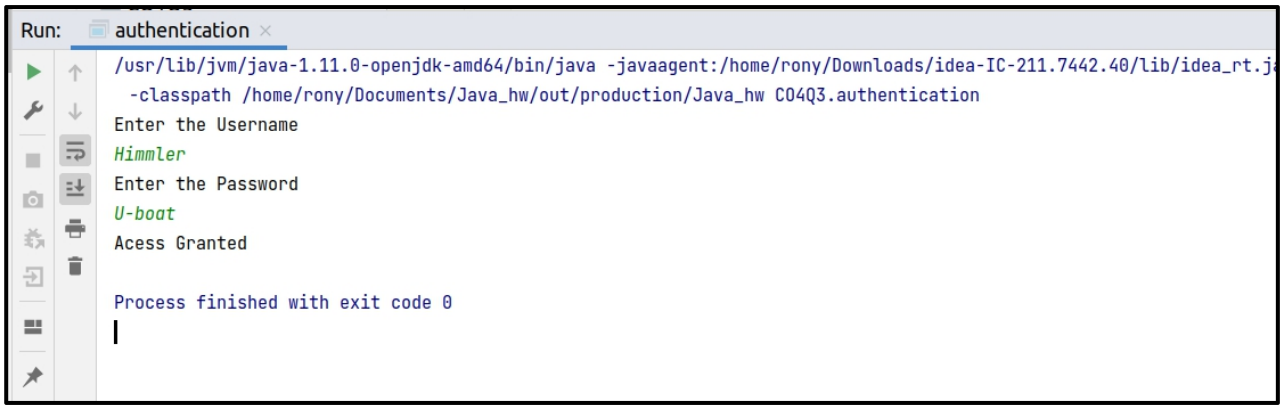
authentication.java	package CO4Q3;
	import java.util.Scanner;
	public class authentication {
	public static void main(String[] args) {
	String username = "Himmler";
	String password = "U-boat";
	Scanner sc = new Scanner(System.in);
	System.out.println("Enter the Username");
	String u1 = sc.nextLine();
	System.out.println("Enter the Password");
	String u2 = sc.nextLine();
	try {

	if ((u1.equals(username)) && (u2.equals(password))) {
	System.out.println("Acess Granted");}
	else {
	throw new credentialexception("Invalid Credentials");} }
	catch (credentialexception e){
	System.out.println(e.getMessage());}}}

credentialexception .java	package CO4Q3;
	public class credentialexception extends Exception{
	public credentialexception(String s){
	super(s);}}

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

OUTPUT :



PROGRAM: 20

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

ALGORITHM :

Step 1: Start

Step 2: Create a class named NegException that inherits Exception class with a constructor inside which we call the Exception class constructor and pass error meaasage.

Step 3: Inside the main(), Read the limit of array

Step 4: Inside the try block,read the array and check if any element is less than 0

Step 5: If true, throw NegException with appropriate message.

Step 6: Calculate the average of the array and print it

Step 7: Inside the catch exception, Print “NEGETIVE EXCEPTION OCCURED”

Step 8:Stop

PROGRAM CODE :

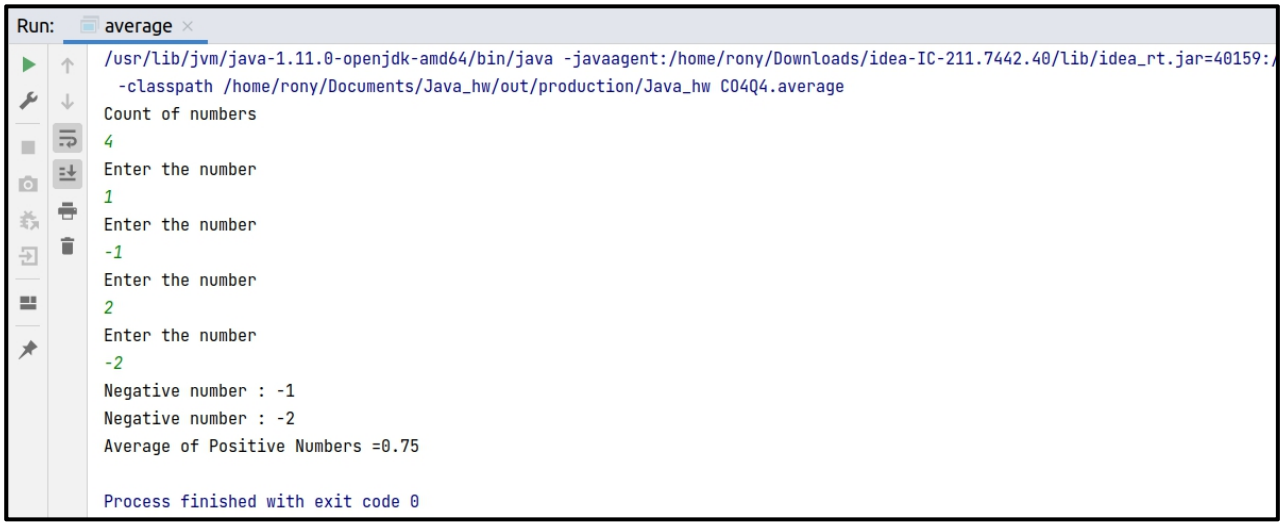
average1.java	package CO4Q4;
	import java.util.Scanner;
	public class average {
	public static void main(String[] args) {
	double sum = 0;
	Scanner sc = new Scanner(System.in);
	System.out.println("Count of numbers");
	int N = sc.nextInt();
	int[] numbers = new int[N];
	for(int i =0;i<N; i++){
	System.out.println("Enter the number");
	numbers[i] = sc.nextInt();
	}
	for(int i =0;i<N; i++){
	try {
	if (numbers[i] >= 0) {
	sum += numbers[i];
	} else {
	throw new negative_exception("Negative number : "
	+numbers[i]);
	}
	}

	catch (negative_exception e){
	System.out.println(e.getMessage());
	}
	}
	double avg = sum/N;
	System.out.println("Average of Positive Numbers ="+avg);
	}
	}

negative_exception.java	package CO4Q4;
	public class negative_exception extends Exception{
	public negative_exception(String s){
	super(s);
	}
	}

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

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)

ALGORITHM :

Step 1: Start

Step 2: Create a class named mul that inherits Thread class with member function as run()

Step 3: Inside run(), Print the multiplication table for 5

Step 4: Create a class named prime that inherits Thread class with memebr function run()

Step 5: Inside run(),Print the prime numbers upto the limit of user's choice

Step 6: Inised the main(), create an object for the classes and call start() using each object

Step 7:Stop

PROGRAM CODE :

driver.java

```
package CO4Q5;

import java.util.Scanner;

public class driver {
   public static void main(String[] args) {
      System.out.println("Enter The number");
      Scanner sc = new Scanner(System.in);
      int number = sc.nextInt();
      mul obj1 = new mul();
      obj1.start();
      prime obj2 = new prime(number);
      obj2.start();
 }
}
```

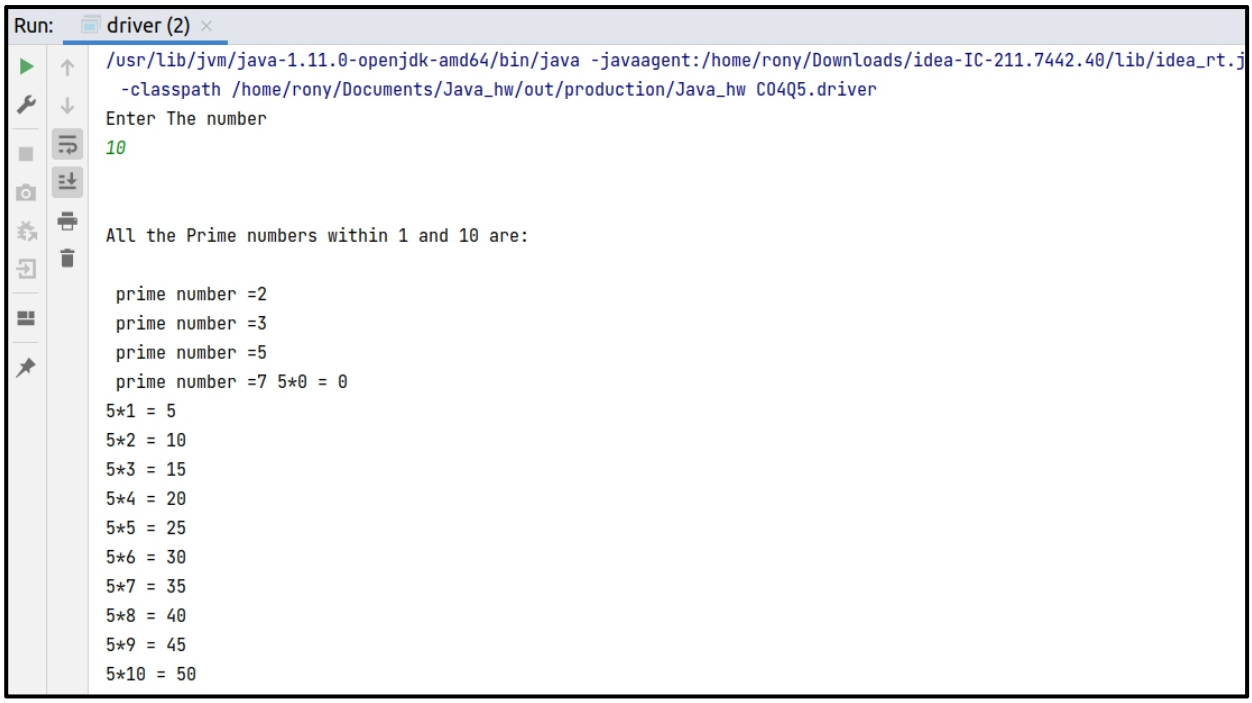
mul.java

```
package CO4Q5;

public class mul extends Thread{
   public void run(){
      System.out.println("\n");
      for(int i =0;i<11;i++){
         System.out.println("5*"+i+" = "+5*i);
      }
}}
```

prime.java	package CO4Q5;
	
	public class prime extends Thread{
	int num;
	public prime(int n){
	this.num=n;
	}
	public void run(){
	int x, y, flg;
	System.out.println("All the Prime numbers within 1 and " + num + " are:");
	for (x = 1; x <= num; x++) {
	if (x == 1 || x == 0)
	continue;
	flg = 1;
	for (y = 2; y <= x / 2; ++y) {
	if (x % y == 0) {
	flg = 0;
	break;
	}
	}
	if (flg == 1)
	System.out.print("\n prime number ="+x + " ");
	}}}

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

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)gs.

ALGORITHM :

Step 1: Start

Step 2: Create a class named even that implements Runnable interface with function run()

Step 3: Inside run(), we read the limit for printing even numbers and print it using for loop.

Step 4:Create another calss fib that implements Runnable interface with function run().

Step 5: Inside run(), Initialise n1 as 0,n2 as 1 and n3 as 0.

Step 6: Check if n<0, if true, print “Enter a positive number” else goto step 7

Step 7: Repeat step8 to 11 until n3>n

Step 8: Print n1

Step 9: n3=n1+n2

Step 10:n1=n2

Step 11: n2=n3

Step 12: Create object e of even and create an object t1 of Thread with its parameterized constructor passing e as parameter

Step 13: Call start() using t1

Step 14: Do the same for class odd with Thread object t2 and call start() using t2

Step 15: Stop

PROGRAM CODE :

driver.java	package CO4Q6;
	import java.util.Scanner;
	public class driver {
	public static void main(String[] args) {
	Scanner sc = new Scanner(System.in);
	System.out.println("Enter the lower limit of range");
	int x = sc.nextInt();
	System.out.println("Enter the Upper limit of range");
	int y = sc.nextInt();
	Runnable r = new fibannoci(x,y);
	Thread obj1 = new Thread(r);
	obj1.start();

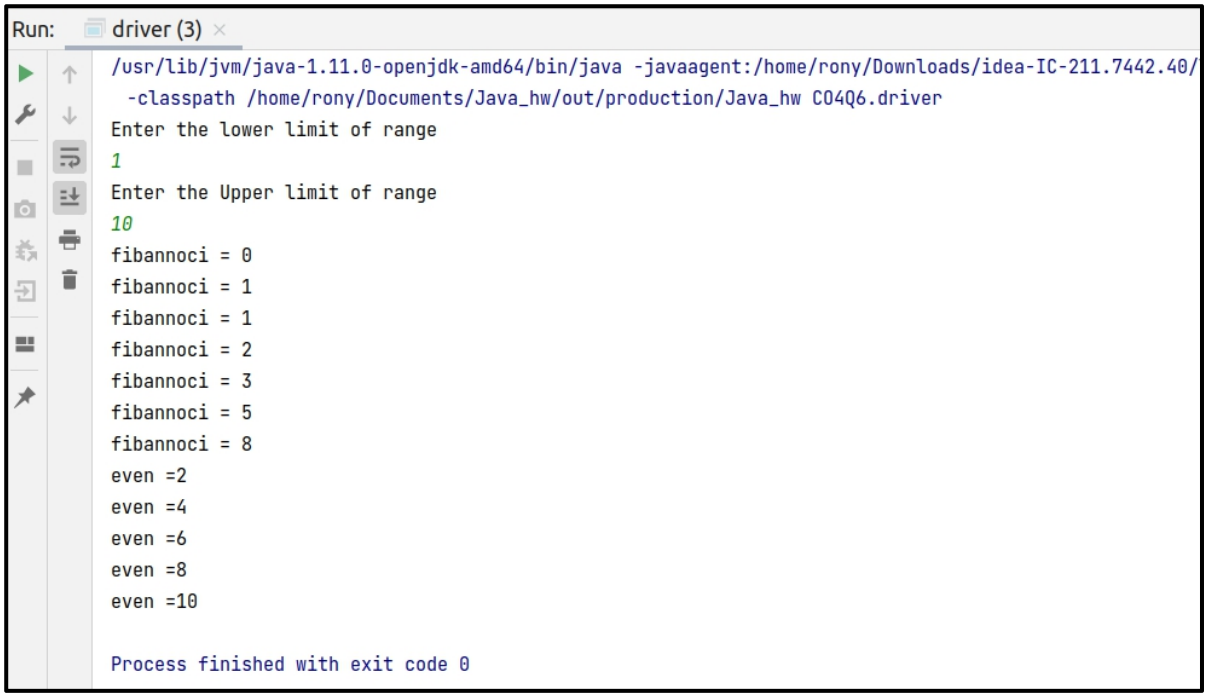
	Runnable p = new even(x,y); Thread obj2 = new Thread(p); obj2.start(); }}

even.java	package CO4Q6;
	public class even implements Runnable{
	int n1;
	int n2;
	even(int x, int y){
	n1 = x;
	n2 = y;
	}
	@Override
	public void run(){
	for(int i= n1; i<=n2; i++){
	if(i%2==0){
	System.out.println("even ="+i);
	}
	}}
	}

fibannoci.java	package CO4Q6;
	public class fibannoci implements Runnable{
	int n1;
	int n2;
	int num =0;
	int x = 0;
	int y = 1;
	fibannoci(int l,int u){
	n1 = l;
	n2 = u;
	}
	@Override
	public void run() {
	System.out.println("fibannoci = "+0);
	System.out.println("fibannoci = "+1);
	while (num < n2-1) {
	num = x + y;
	if ((num >= n1)&&(num<=n2)) {
	System.out.println("fibannoci = "+num);
	}
	x = y;
	y = num;
	}}}

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

OUTPUT :



PROGRAM: 23

AIM : . Producer/Consumer using ITC

ALGORITHM :

Step 1: Start

Step 2:In PC class (A class that has both produce and consume methods), a linked list of jobs and a capacity of the list is added to check that producer does not produce if the list is full.

Step 3:In Producer class, the value is initialized as 0.

Step 4:We have an infinite outer loop to insert values in the list. Inside this loop, we have a synchronized block so that only a producer or a consumer thread runs at a time. An inner loop is there before adding the jobs to list that checks if the job list is full, the producer thread gives up the intrinsic lock on PC and goes on the waiting state.

Step 5:If the list is empty, the control passes to below the loop and it adds a value in the list.

Step 6:In the Consumer class, we again have an infinite loop to extract a value from the list. Inside, we also have an inner loop which checks if the list is empty.

Step 7:If it is empty then we make the consumer thread give up the lock on PC and passes the control to producer thread for producing more jobs.

Step 8:If the list is not empty, we go round the loop and removes an item from the list.

Step 9:In both the methods, we use notify at the end of all statements. The reason is simple, once you have something in list, you can have the consumer thread consume it, or if you have consumed something, you can have the producer produce something.

Step 10:sleep() at the end of both methods just make the output of program run in step wise manner and not display everything all at once so that you can see what actually is happening in the program.

Step 11:Stop

PROGRAM CODE :

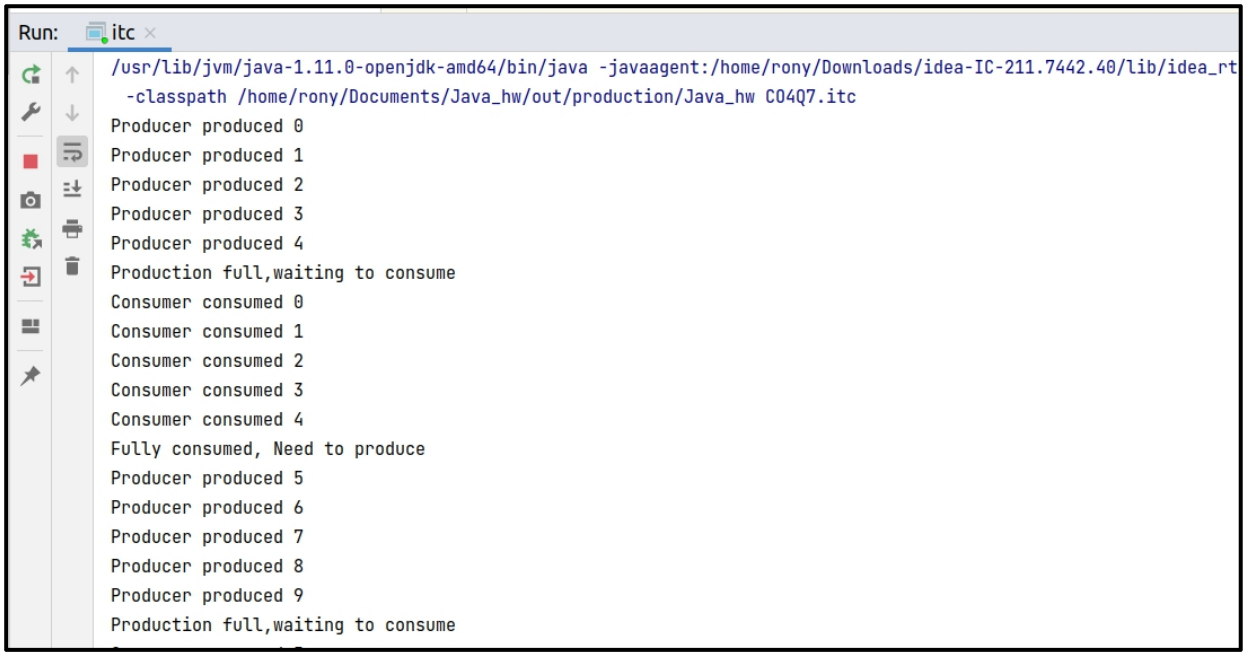
itc.java	package CO4Q7;
	//Producer/Consumer using ITC
	import java.util.ArrayList;
	import java.util.List;
	
	class Producer implements Runnable
	{
	List<Integer> flist;
	int max_size = 5;

```
    int i=0;
    Producer(List<Integer> flist)
    {
       this.flist = flist;
    }
    @Override
    public void run()
    {
       while(true)
       {
          try
          {
             produce(i++);
          } catch (Exception e)
          {
             System.out.println("Interuption "+e);
          }
       }
    }
    public void produce(int i) throws InterruptedException
    {
       synchronized (flist)
       {
          while(flist.size()==max_size)
          {
             System.out.println("Production full,waiting to consume");
             flist.wait();
          }
       }
       synchronized(flist)
       {
          System.out.println("Producer produced "+i);
          flist.add(i);
          flist.notify();
       }
    }

}
class Consumer implements Runnable
{
    List<Integer> flist;
    Consumer(List<Integer> flist)
    {
       this.flist = flist;
    }

    @Override
    public void run()
    {
       while(true)
```

```
        {
            try
            {
                consume();
            } catch (Exception e)
            {
                System.out.println("Exception "+e);
            }
        }
    }
    public void consume() throws InterruptedException
    {
        synchronized (flist)
        {
            while(flist.isEmpty())
            {
                System.out.println("Fully consumed, Need to produce");
                flist.notify();
                Thread.sleep(500);
                flist.wait();
            }
        }
        synchronized(flist)
        {
            Thread.sleep(1000);
            System.out.println("Consumer consumed "+flist.remove(0));
        }
    }

}
public class itc {
    public static void main(String[] args)
    {
        List<Integer> flist = new ArrayList<Integer>();
        Thread th1 = new Thread(new Producer(flist));
        Thread th2 = new Thread(new Consumer(flist));
        th1.start();
        th2.start();
    }
}
```

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

OUTPUT :



PROGRAM: 24

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

ALGORITHM :

Step 1: Start

Step 2: Create a class named stack with data members as a(an array),top(set as -1),ch,item,i; a function named menu()

Step 3: Inside menu(), give choices to push,pop and display the stack

Step 4: If the choice is 1, then check whether the stack is full, else add an element into the stack.

Step 5: If the choice is 2, then check whether the stack is empty, else delete an element into the stack.

Step 6: If the choice is 3, then check whether the stack is empty, else print all the elements in the stack.

Step 7: If the choice is greater than 4, then print “Invalid option”.

Step 8: Inside the main(), create an object of type stack and call the menu() function.

Step 9:Stop

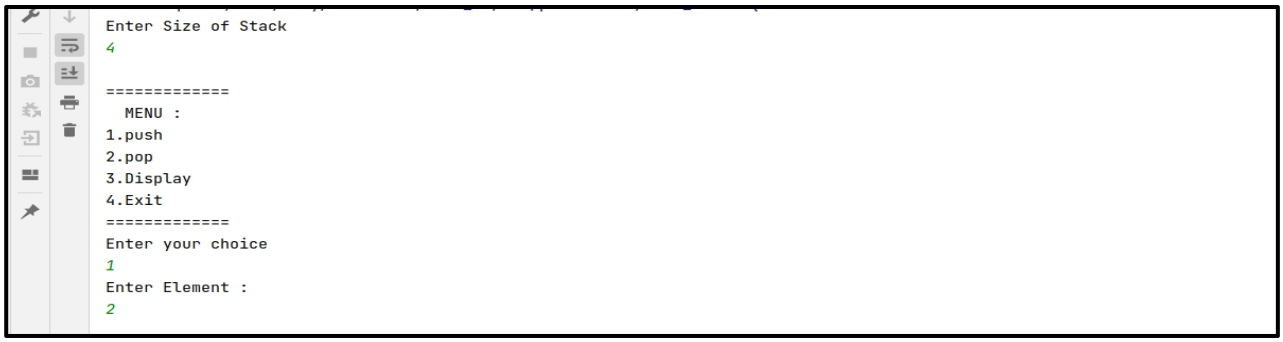
PROGRAM CODE :

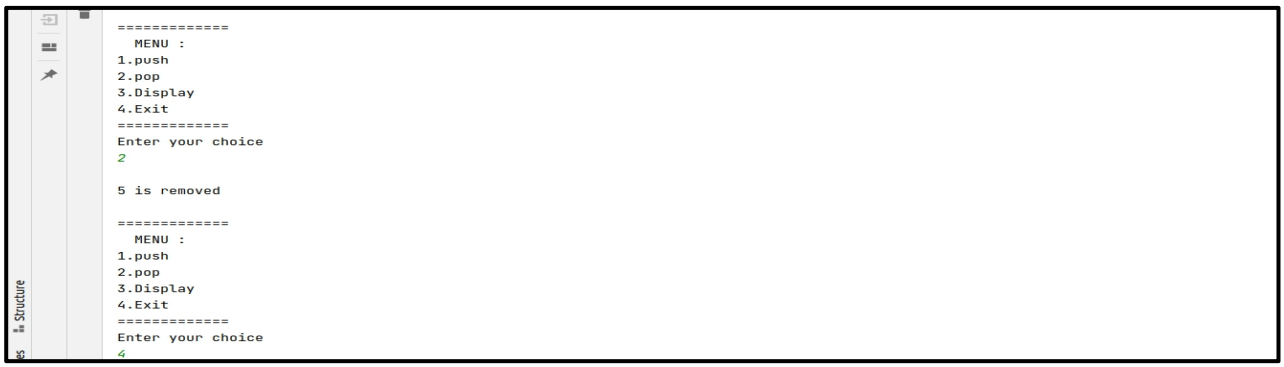
driver.java	package CO4Q8;
	import java.util.*;
	class operations{
	public void operation()
	{
	int top =-1,ch,n,e;
	Scanner inp = new Scanner(System.in);
	System.out.println("Enter Size of Stack");
	n = inp.nextInt();
	int size=n-1;
	int[] arr = new int[n];
	
	do {
	System.out.println("\n==========\n MENU : \n1.push \n2.pop \n3.Display \n4.Exit \n==========");
	System.out.println("Enter your choice");
	ch = inp.nextInt();
	switch(ch)
	{
	case 1 :

```
                if(top == size)
                {
                   System.out.println(" ***  Stack is Full *** ");}
                else
                {
                   System.out.println("Enter Element : ");
                   e = inp.nextInt();
                   top++;
                   arr[top] =e; }
                break;
             case 2 :
                if(top == -1)
                {
                   System.out.println("\n***  Stack is empty *** ");
                }
                else
                {
                   System.out.println("\n"+ arr[top] + " is removed ");
                   top--;}
                break;
             case 3 :
                if(top == -1)
                {
                   System.out.println(" ***  Stack is empty ***");
                }
                else
                {
                   System.out.println("\n***  Stack : ***\n");
                   for(int i=top;i>=0;i--)
                   {
                      System.out.println("  " +arr[i]);
                      System.out.println("-----");
                   }}
                break;
             case 4 :
                System.exit(0);
             default : System.out.println("Invalid Choice");
          }
       }while(ch !=4);

    }}
public class driver {
   public static void main(String[] args) {
      operations obj = new operations();
      obj.operation();
   }
}
```

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

OUTPUT :





PROGRAM: 25

AIM : . Using generic method perform Bubble sort.

ALGORITHM :

Step 1: Start

Step 2: Create a function named bubblesort(array)

Step 3: n<- length of array

Step 4: Intialize temp<-0

Step 5: i<-0

Step 6:Reapeat steps from to until i>n

Step 7: j<-1,repeat the steps from to until j>n-I

Step 8: check if array[i] >array[j], if true,swap them;else increment j

Step 9: Inside main () Initialize an array with elements and the print the same

Step 10: Call the function bubblesort() and pass the array as parameter

Step 11: Print the sorted array

Step 12: Stop

PROGRAM CODE :

driver.java

```
package CO2Q1;
//. Program to Sort strings

import java.util.Scanner;

import java.util.Arrays;

public class CO2Q1 {

    public static void main(String[] args) {
        int i,j;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of words");

        int num=sc.nextInt();

        String word[]=new String[num];

        sc.nextLine();
```

```
        for( i=0;i<num;i++){
           System.out.println("\nEnter a Word\n");
           word[i]=sc.nextLine();
        }
        for( i=0;i<num-1;i++){
           for( j=i+1;j<num;j++){
              if(word[i].compareTo(word[j])>0){
                 String temp = word[i];
                 word[i]=word[j];
                 word[j]=temp;
              }
           } }
        System.out.println("Sorted Strings using compareTo function
="+Arrays.toString(word));
        System.out.println(word);
     }
     }
```

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

OUTPUT :

PROGRAM: 26

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

ALGORITHM :

Step 1: Start

Step 2: Create an object ‘a’ of type ArrayList.

Step 3: Put values into it using add()

Step 4: Manipulate the list using built in functions.

Step 5: Print the elements in a

Step 6: Stop

PROGRAM CODE :

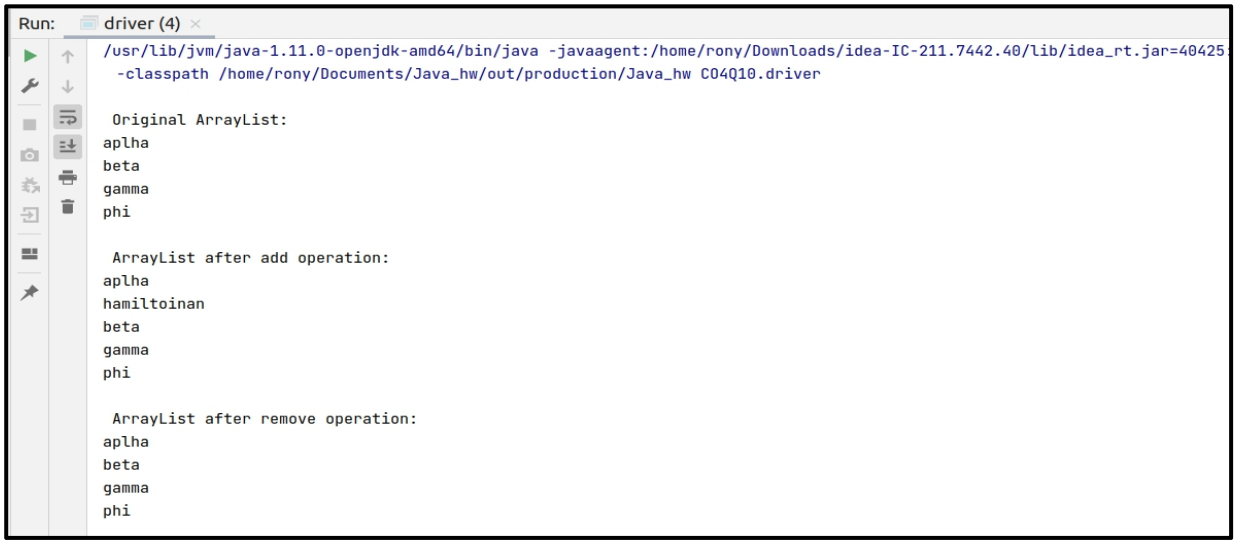
CO2Q1.java

```
package CO4Q10;
import java.util.*;
public class driver {
   public static void main(String[] args) {
      ArrayList<String> obj = new ArrayList<String>();
      obj.add("aplha");
      obj.add("beta");
      obj.add("gamma");
      obj.add("phi");
      System.out.println("\n Original ArrayList:");
      for(String str:obj)
         System.out.println(str);
      obj.add(1, "hamiltoinan");
      System.out.println("\n ArrayList after add operation:");
      for(String str:obj)
         System.out.println(str);
      obj.remove("hamiltoinan");
      System.out.println("\n ArrayList after remove operation:");
      for(String str:obj)
         System.out.println(str);
      obj.remove(3);
      System.out.println("\n Final ArrayList:");
      for(String str:obj)
         System.out.println(str);
      Collections.sort(obj);
      System.out.println("\n  ArrayList after sorting:");
      for (String str : obj)
         System.out.println(str);
```

```
        System.out.println("\n Object at index 2:"+obj.get(2));
        System.out.println("\n Six is in the ArrayList :"+obj.contains("degree"));
        System.out.println("\n Two is in the ArrayList :"+obj.contains("dell"));
        System.out.println("\n Size of the ArrayList:"+obj.size());
        obj.clear();
        System.out.println("\n** ArrayList Removed **");
    }
}
```

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

OUTPUT :



PROGRAM: 27

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

ALGORITHM :

Step 1:Start

Step 2: Create an object of the class linkedlist.

Step 3: Adding elements to the linked list using method add().

Step 4: Remove all the elements of linkedlist using method clear().

Step 5:Display linkedlist.

Step 6: Stop

PROGRAM CODE :

q11driver.java

```
package CO4Q11;
import java.util.*;
public class q11driver {
    public static  void main(String[] args){
        LinkedList<String> L=new LinkedList<>();
        L.add("Gold");
        L.add("Silver");
        L.add("Bronze");
        L.add(0,"Olympics Medals");
        System.out.println(L);
        L.remove("Bronze");
        System.out.println(L);
        L.remove(2);
        System.out.println(L);
        L.removeLast();
        System.out.println(L);
        L.removeFirst();
        System.out.println(L);
    }
}
```

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

OUTPUT :

PROGRAM: 28

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

ALGORITHM :

Step 1:Start

Step 2: Create an object of the class Stack.

Step 3: Adding elements to the stack using method add().

Step 4: Remove the element of stack at position 'pos' using method remove(pos).

Step 5:Display removed element and Stack.

Step 6: Stop

PROGRAM CODE :

q12driver.java	package CO4Q12;
	
	import java.util.*;
	
	public class q12driver {
	
	public static void main(String[] args) {
	
	Stack<Integer> st = new Stack<>();
	st.push(11);
	st.push(22);
	st.push(33);
	st.push(44);
	
	System.out.println("Stack = "+st);
	Scanner sc = new Scanner(System.in);
	System.out.println("Enter the position : ");
	int x = sc.nextInt();
	st.remove(x);
	System.out.println("Stack = "+st);
	
	}

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

OUTPUT :

PROGRAM: 29

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

ALGORITHM :

Step 1:Start

Step 2: Create an object of the class PriorityQueue.

Step 3: Adding elements to the PriorityQueue using method add().

Step 5:Display PriorityQueue.

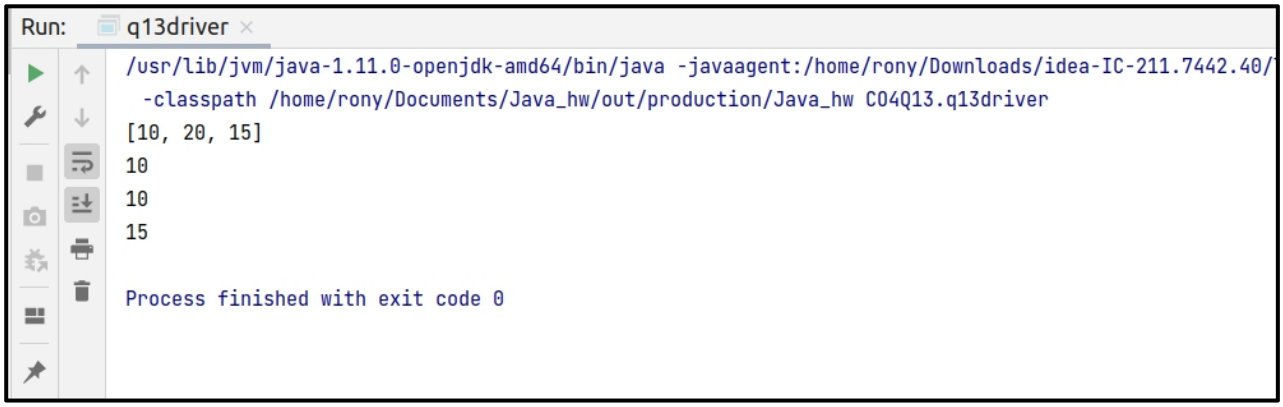
Step 6: Stop

PROGRAM CODE :

q13driver.java	package CO4Q13;
	import java.util.PriorityQueue;
	public class q13driver {
	public static void main(String[] args) {
	PriorityQueue <Integer> pq = new PriorityQueue<>();
	pq.add(10);
	pq.add(20);
	pq.add(15);
	System.out.println(pq);
	System.out.println(pq.peek());
	System.out.println(pq.poll());
	System.out.println(pq.peek());
	}}

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

OUTPUT :



PROGRAM: 30

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

ALGORITHM :

Step 1:Start

Step 2: Create an object of the class ArrayDeque.

Step 3: Adding elements to the queue using method add().

Step 4: Removing elements of queue using method pop().

Step 5:Display Queue.

Step 6: Stop

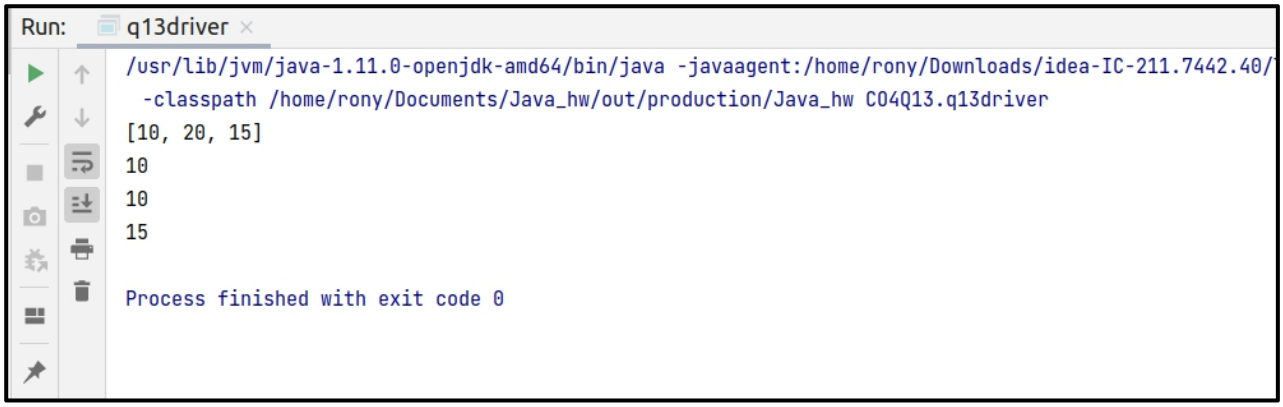
PROGRAM CODE :

q14driver.java

```
package CO4Q14;
import java.util.*;
public class q14driver {
    public static void main(String[] args) {
        Deque<Integer> dq = new ArrayDeque<>();
        dq.add(1);
        dq.add(2);
        dq.add(3);
        System.out.println("Inserting three elements : ");
        for (Integer integer : dq) {
            System.out.println(integer);
        }
        dq.pop();
        System.out.println("After popping : ");
        for (Integer integer : dq) {
            System.out.println(integer);
        }
        dq.remove(3);
        System.out.println("Removing the element 3 :"+dq);
    }
}
```

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

OUTPUT :



PROGRAM: 31

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

ALGORITHM :

Step 1:Start

Step 2: Create an object of the class LinkedHashset.

Step 3: Adding elements to the HashSet using method add().

Step 4:Display LinkedHashSet.

Step 6: Stop

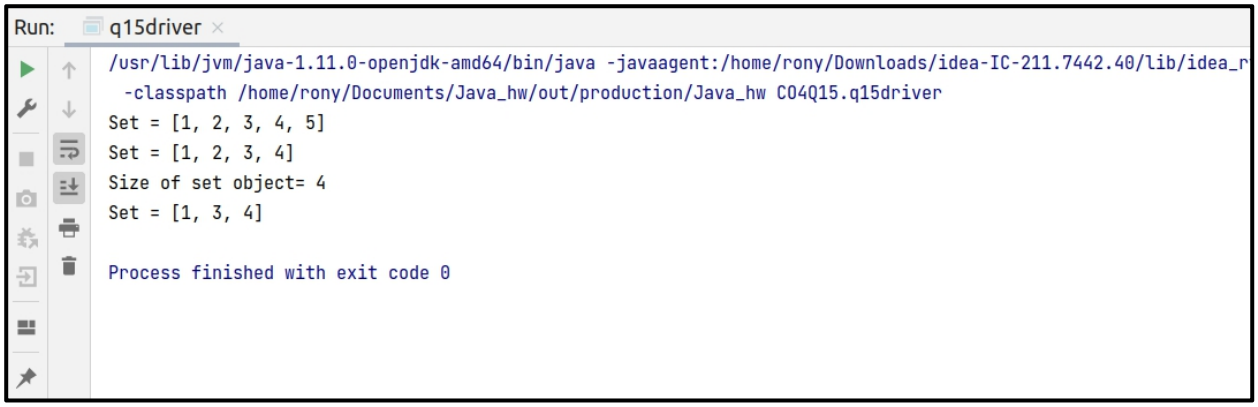
PROGRAM CODE :

q15driver.java

```
package CO4Q15;

import java.util.*;

public class q15driver {
   public static void main(String[] args) {
      LinkedHashSet <Integer> HS = new LinkedHashSet<>();
      HS.add(1);
      HS.add(2);
      HS.add(3);
      HS.add(4);
      HS.add(5);
      System.out.println("Set = "+HS);
      HS.remove(5);
      System.out.println("Set = "+HS);
      int x=HS.size();
      System.out.println("Size of set object= "+x);
      HS.remove(2);
      System.out.println("Set = "+HS);

   }
}
```

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

OUTPUT :



PROGRAM: 32

AIM : Write a Java program to compare two hash set

ALGORITHM :

Step 1:Start

Step 2: Create two objects of the class LinkedHashset.

Step 3: Adding elements to the two objects of LinkedHashSet using method add().

Step 4:Checking weather elements of first Hashset is present in second Hashset.

Step 5:If Step 4 is true print Yes,else print No.

Step 6: Stop

PROGRAM CODE :

q16driver.java

```
package CO4Q16;
import java.util.HashSet;
public class q16driver {
   public static void main(String[] args) {
      // Create a empty hash set
      HashSet<String> h_set = new HashSet<String>();
      // use add() method to add values in the hash set
      h_set.add("Red");
      h_set.add("Green");
      h_set.add("Black");
      h_set.add("Orange");
      h_set.add("Pink");

      HashSet<String>h_set2 = new HashSet<String>();
      h_set2.add("Red");
      h_set2.add("Pink");
      h_set2.add("Black");
      h_set2.add("Orange");
      //comparison output in hash set
      for (String element : h_set){
         System.out.println(h_set2.contains(element) ? "Yes" : "No");
      }
   }
}
```

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

OUTPUT :

PROGRAM: 33

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

ALGORITHM :

Step 1:Start

Step 2:Initialization of a Map using Generics.

Step 3:Adding values into map using method put() and display.

Step 4:Updating values using method put() by mentioning index of value and display.

Step 5:Removing values from map using method remove() and display.

Step 6: Stop

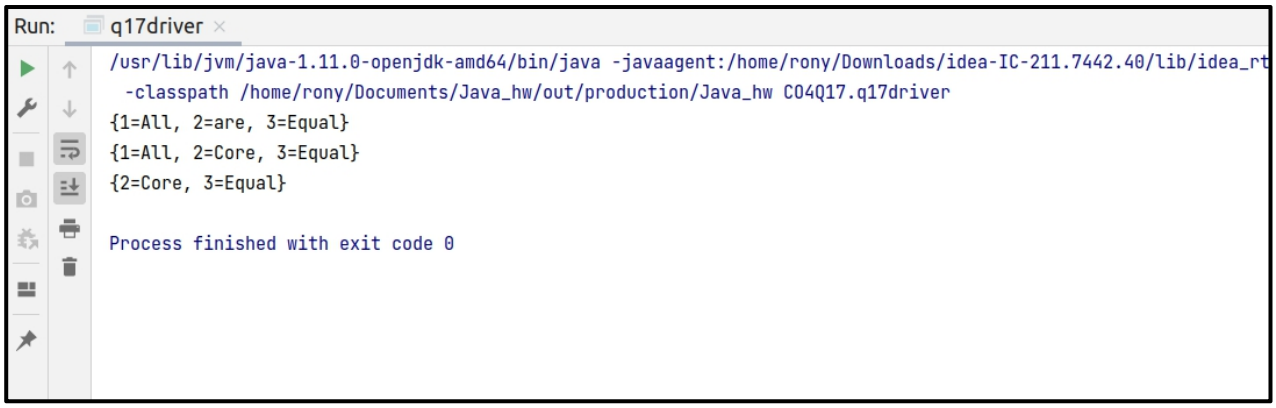
PROGRAM CODE :

q17driver.java

```
package CO4Q17;

import java.util.*;

public class q17driver {

   public static void main(String args[])
   {
      Map<Integer, String> hm2= new HashMap<Integer, String>();
      hm2.put(new Integer(1), "All");
      hm2.put(new Integer(2), "are");

      hm2.put(new Integer(3), "Equal");

      System.out.println(hm2);
      hm2.put(new Integer(2), "Core");
      System.out.println(hm2);

      hm2.remove(new Integer(1));

      System.out.println(hm2);
   }

}
```

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

OUTPUT :



PROGRAM: 34

AIM : Program to Convert HashMap to TreeMap

ALGORITHM :

Step 1:Get the HashMap to be converted.

Step 2:Create a new TreeMap

Step 3:Pass the hashMap to putAll() method of treeMap

Step 4:Return the formed TreeMap

Step 6: Stop

PROGRAM CODE :

q18driver.java

```
package CO4Q18;
import java.util.*;
public class q18driver {

    public static void main(String args[]) {

        Map<String, String> map = new HashMap<>();
        System.out.println("Enter the limit:");
        Scanner inp = new Scanner(System.in);
        int n= inp.nextInt();
        System.out.println("Enter the Roll number and Name");
        while(n!=0) {

            String e= inp.next();
            String s= inp.next();
            map.put(e, s);
            n--;
        }
        System.out.println("HashMap:"+map);
        Map<String, String> treeMap = new TreeMap<>();
        treeMap.putAll(map);
        System.out.println("TreeMap:"+treeMap);
    }
}
```

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

OUTPUT :