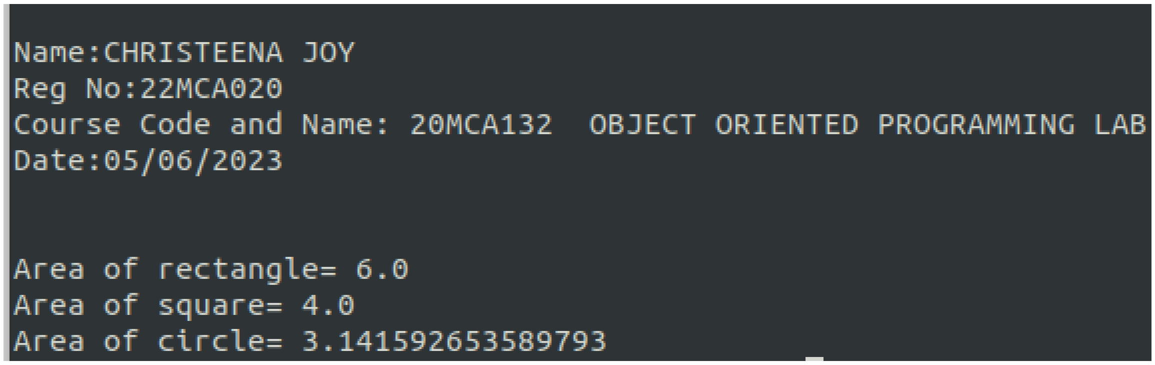
Cycle 3

1. Area of different shapes using overloaded functions.

CODE:

```
public class AreaCalculator{

public static double calculateArea(double l,double b){
double area;
area=l*b;
return area;
}

public static double calculateArea(float r){
double area;
area=Math.PI*r;
return area;
}

public static double calculateArea(int l){
double area;
area=l*l;
return area;
}

public static void main(String args[]){
System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and Name:
20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:05/06/2023\n\n");
int l=2,b=3;
float r=1.0f;
System.out.println("Area of rectangle= "+calculateArea(l,b));
System.out.println("Area of square= "+calculateArea(l));
System.out.println("Area of circle= "+calculateArea(r));
}
}
```

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:05/06/2023


Area of rectangle= 6.0
Area of square= 4.0
Area of circle= 3.141592653589793
```

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

CODE:

Employee.java

```
public class Employee{
int Empid;
String Name;
int Salary;
String address;

public Employee(int e,String n,int s,String a){
this.Empid=e;
this.Name=n;
this.Salary=s;
this.address=a;
}

public int getEmpid(){
return Empid;
}

public String getName(){
return Name;
}

public int getSalary(){
return Salary;
}

public String getAddress(){
```

```
return address;
}
}
```

Teacher.java

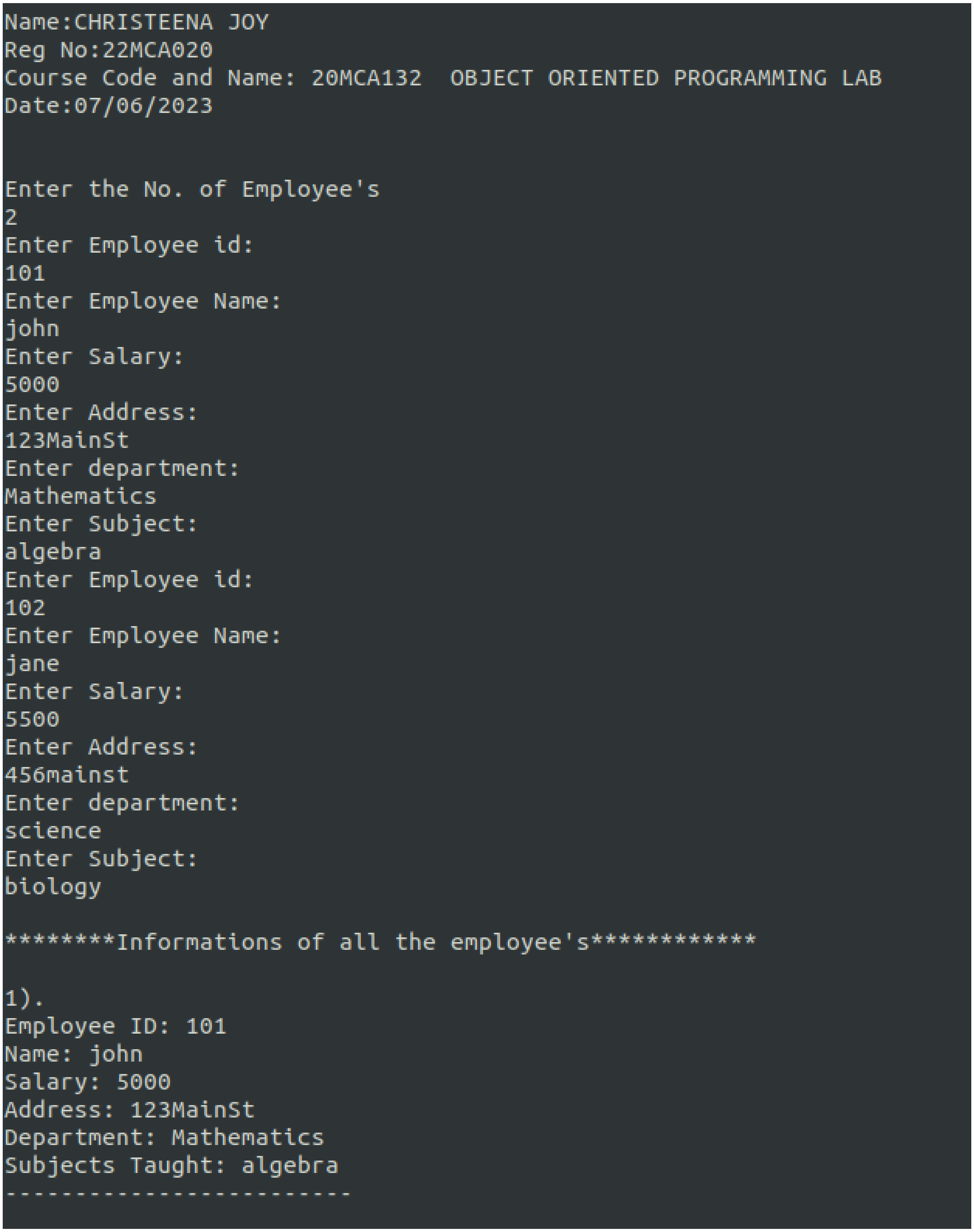
```
public class Teacher extends Employee{
String department,Subjects;
public Teacher(int Empid,String name,int Salary,String a,String d,String s){
super(Empid,name,Salary,a);
this.department=d;
this.Subjects=s;
}

public String getDepartment(){
return department;
}

public String getSubjects(){
return Subjects;
}

 public void display() {
 System.out.println("Employee ID: " + getEmpid());
 System.out.println("Name: " + getName());
 System.out.println("Salary: " + getSalary());
 System.out.println("Address: " + getAddress());
 System.out.println("Department: " + department);
 System.out.println("Subjects Taught: " + Subjects);
 System.out.println("-------------------------");
    }

}
```

Employee_main.java

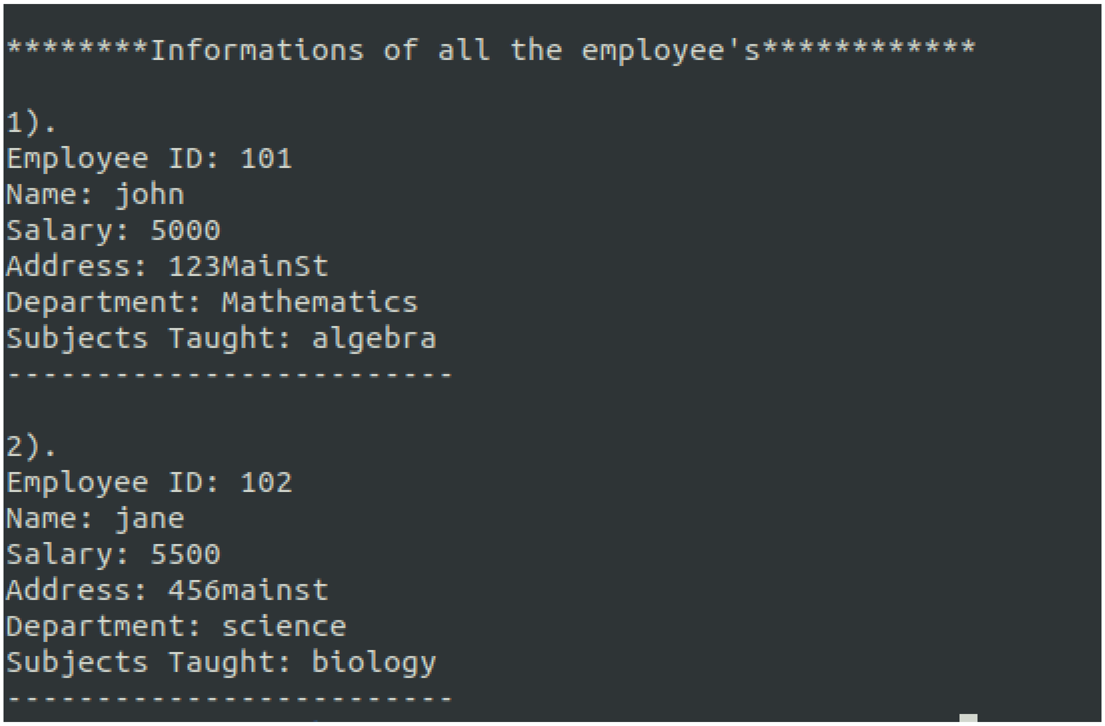
```
import java.util.Scanner;
public class Employee_main{
public static void main(String[] args) {
System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and Name:
20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:07/06/2023\n\n");
System.out.println("Enter the No. of Employee's");
   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: ");
      int Salary=sc.nextInt();
      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);



   }
   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();
 }

}
```

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:07/06/2023


Enter the No. of Employee's
2
Enter Employee id:
101
Enter Employee Name:
john
Enter Salary:
5000
Enter Address:
123MainSt
Enter department:
Mathematics
Enter Subject:
algebra
Enter Employee id:
102
Enter Employee Name:
jane
Enter Salary:
5500
Enter Address:
456mainst
Enter department:
science
Enter Subject:
biology

********Informations of all the employee's************

1).
Employee ID: 101
Name: john
Salary: 5000
Address: 123MainSt
Department: Mathematics
Subjects Taught: algebra
-------------------------
```



```
********Informations of all the employee's************

1).
Employee ID: 101
Name: john
Salary: 5000
Address: 123MainSt
Department: Mathematics
Subjects Taught: algebra
-------------------------

2).
Employee ID: 102
Name: jane
Salary: 5500
Address: 456mainst
Department: science
Subjects Taught: biology
-------------------------
```

3. 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.

CODE:

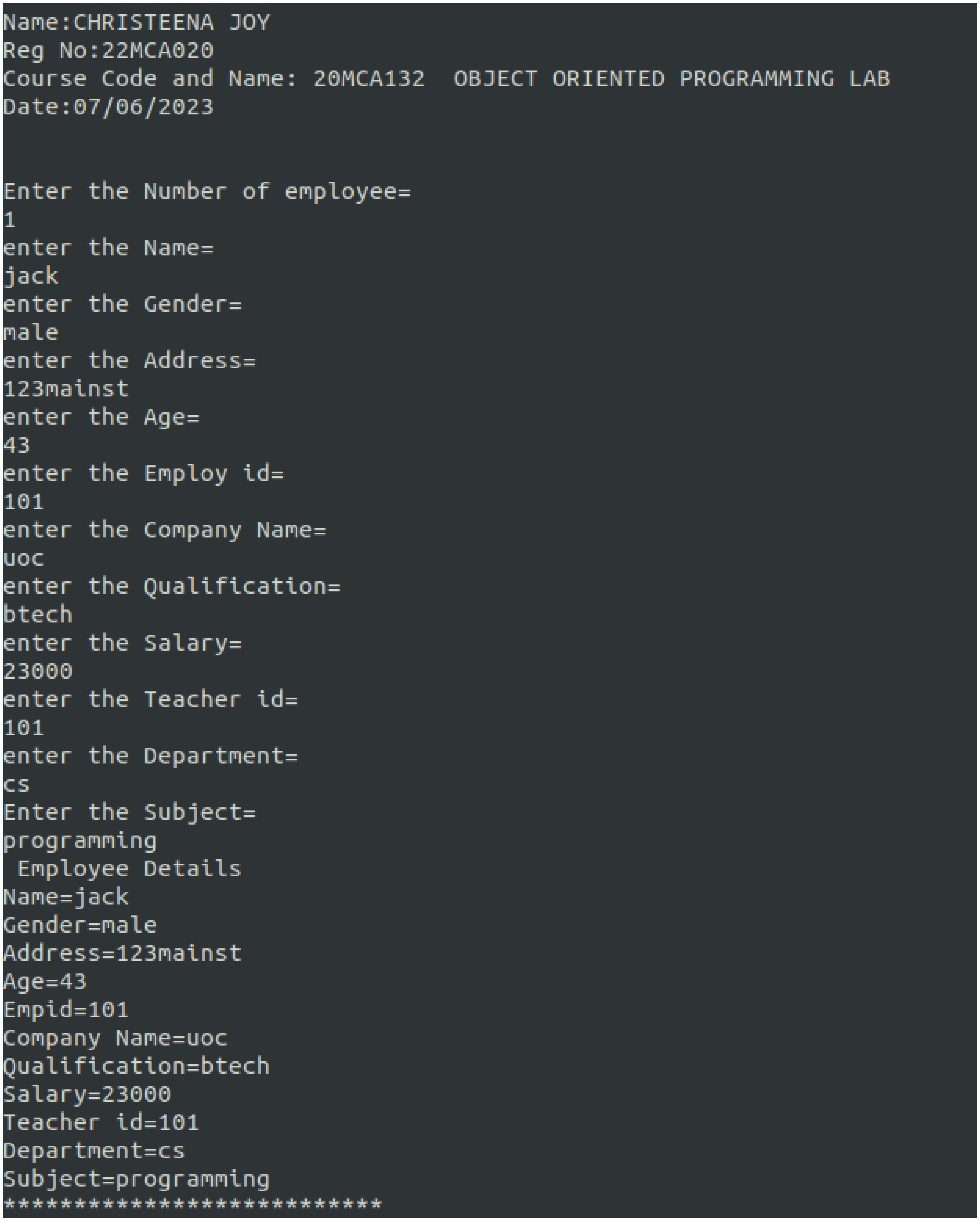
```
import java.util.*;
class Person{
String Name;
String Gender;
String Address;
String Age;
public Person(String Name,String Gender,String Address,String Age) {
this.Name=Name;
this.Gender=Gender;
this.Address=Address;
this.Age=Age;
}
}
class Employee extends Person
{
String Empid;
String Company_Name;
String Qualification;
String Salary;
public Employee(String Name,String Gender,String Address,String Age ,String Empid,String
Company_Name, String Qualification,String Salary)
{
super(Name,Gender,Address,Age);
this.Empid= Empid;
this.Company_Name=Company_Name;
this.Qualification=Qualification;
this.Salary=Salary;
}
}
class Teacher extends Employee
{
String Teacherid;
String Department;
String Subject;
public Teacher(String Name,String Gender,String Address,String Age,String Empid,String
Company_Name,String Qualification,String Salary,String Teacherid, String Department,String
Subject)
{
super(Name,Gender,Address,Age,Empid,Name,Qualification, Salary);
this.Teacherid=Teacherid;
this.Department=Department;
this.Subject=Subject;
}
public void read()
{ Scanner in =new Scanner(System.in);
System.out.println("enter the Name=");
Name=in.nextLine();
System.out.println("enter the Gender=");
Gender=in.nextLine();
System.out.println("enter the Address=");
Address=in.nextLine();
System.out.println("enter the Age=");
Age=in.nextLine();
System.out.println("enter the Employ id=");
Empid=in.nextLine();
System.out.println("enter the Company Name=");
Company_Name=in.nextLine();
System.out.println("enter the Qualification=");
```

```
Qualification=in.nextLine();
System.out.println("enter the Salary=");
Salary=in.nextLine();
System.out.println("enter the Teacher id=");
Teacherid=in.nextLine();
System.out.println("enter the Department=");
Department=in.nextLine();
System.out.println("Enter the Subject=");
Subject=in.nextLine();
}
public void display()
{ System.out.println(" Employee Details ");
System.out.println("Name="+ Name);
System.out.println("Gender=" + Gender);
System.out.println("Address=" + Address);
System.out.println("Age=" + Age);
System.out.println("Empid=" + Empid);
System.out.println("Company Name=" + Company_Name);
System.out.println("Qualification=" + Qualification);
System.out.println("Salary=" + Salary);
System.out.println("Teacher id=" + Teacherid);
System.out.println("Department=" + Department);
System.out.println("Subject=" + Subject);
System.out.println("***************************"); }
}
public class PersonMain
{
public static void main(String Args[])
{ System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:07/06/2023\n\n");
int i,n;
Scanner in =new Scanner(System.in);
System.out.println("Enter the Number of employee=");
n=in.nextInt();
```

```
Teacher T[] = new Teacher[n];
for(i=0;i<n;i++)
{
T[i]=new
Teacher("Name","Gender","Address","Age","Empid","Name","Qualification","Salary","
Teacherid","Department","Subject");
T[i].read();
}
for(i=0;i<n;i++)
{
T[i].display();
}
}
}
```

OUTPUT:

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:07/06/2023


Enter the Number of employee=
1
enter the Name=
jack
enter the Gender=
male
enter the Address=
123mainst
enter the Age=
43
enter the Employ id=
101
enter the Company Name=
uoc
enter the Qualification=
btech
enter the Salary=
23000
enter the Teacher id=
101
enter the Department=
cs
Enter the Subject=
programming
 Employee Details
Name=jack
Gender=male
Address=123mainst
Age=43
Empid=101
Company Name=uoc
Qualification=btech
Salary=23000
Teacher id=101
Department=cs
Subject=programming
***************************
```

4. 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.

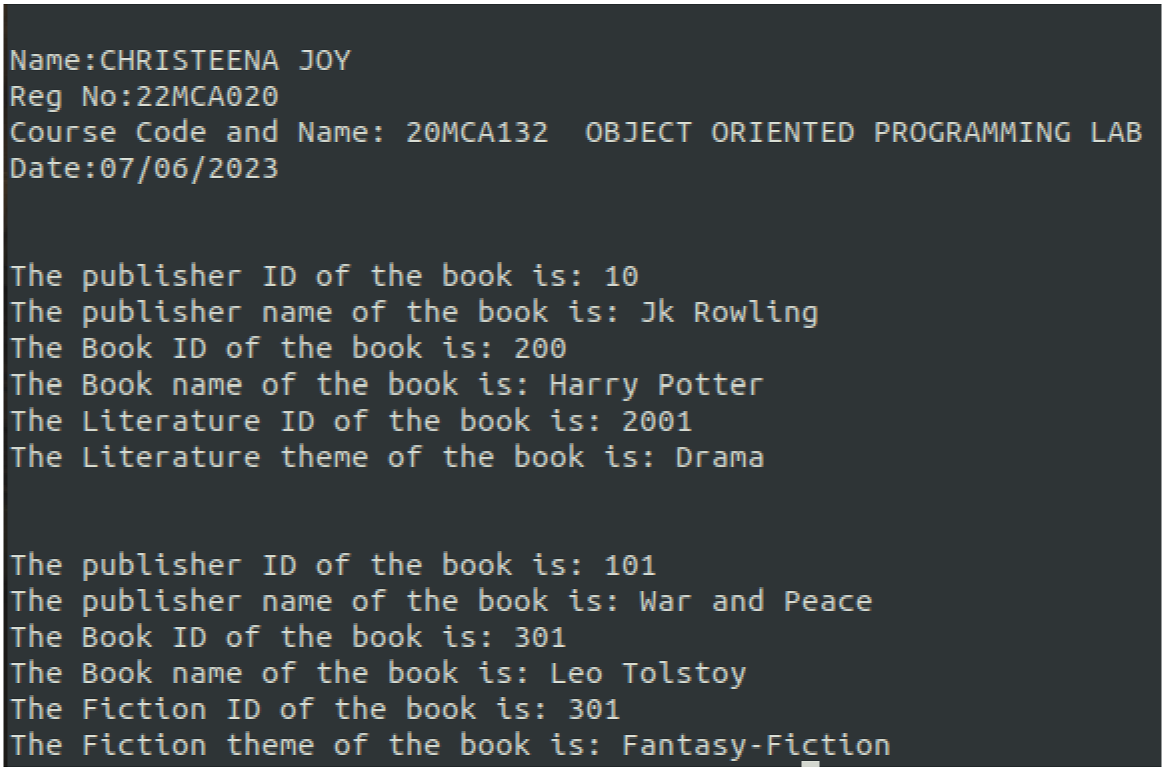
CODE:

```
import java.util.Scanner;
class Publisher{
int publisher_id;
String publisher_name;
Publisher(int publisher_id, String publisher_name){
this.publisher_id= publisher_id;
this.publisher_name= publisher_name;
}
}
class Book extends Publisher{
int book_id;
String book_name;
Book(int publisher_id, String publisher_name, int book_id, String book_name) {
super(publisher_id, publisher_name);
this.book_id= book_id;
this.book_name= book_name;
}
}
class Literature extends Book{
int literature_id;
String literature_theme;
Literature(int publisher_id, String publisher_name, int book_id, String book_name, int
literature_id, String literature_theme) {
super(publisher_id, publisher_name, book_id, book_name);
this.literature_id= literature_id;
this.literature_theme= literature_theme;
```

```
}
void displayDetails() {
System.out.println("The publisher ID of the book is: " + this.publisher_id);
System.out.println("The publisher name of the book is: " + this.publisher_name);
System.out.println("The Book ID of the book is: " + this.book_id); System.out.println("The Book
name of the book is: " + this.book_name); System.out.println("The Literature ID of the book is: "
+ this.literature_id); System.out.println("The Literature theme of the book is: " +
this.literature_theme); }
}
class Fiction extends Book{
int fiction_id;
String fiction_theme;
Fiction(int publisher_id, String publisher_name, int book_id, String book_name, int fiction_id,
String fiction_theme) {
super(publisher_id, publisher_name, book_id, book_name);
this.fiction_id= fiction_id;
this.fiction_theme= fiction_theme;
}
void displayDetails() {
System.out.println("The publisher ID of the book is: " + this.publisher_id);
System.out.println("The publisher name of the book is: " + this.publisher_name);
System.out.println("The Book ID of the book is: " + this.book_id); System.out.println("The Book
name of the book is: " + this.book_name); System.out.println("The Fiction ID of the book is: " +
this.fiction_id); System.out.println("The Fiction theme of the book is: " + this.fiction_theme);
}
}
public class BookMain {
public static void main(String[] args) {
System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and Name:
20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:07/06/2023\n\n");
Literature literature= new Literature(10,"Jk Rowling",200,"Harry Potter",2001,"Drama");
Fiction fiction= new Fiction(101, "War and Peace", 301, "Leo Tolstoy", 301, "Fantasy-Fiction");
literature.displayDetails();
System.out.println("\n");
```

```
fiction.displayDetails();
}
}
```

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:07/06/2023


The publisher ID of the book is: 10
The publisher name of the book is: Jk Rowling
The Book ID of the book is: 200
The Book name of the book is: Harry Potter
The Literature ID of the book is: 2001
The Literature theme of the book is: Drama


The publisher ID of the book is: 101
The publisher name of the book is: War and Peace
The Book ID of the book is: 301
The Book name of the book is: Leo Tolstoy
The Fiction ID of the book is: 301
The Fiction theme of the book is: Fantasy-Fiction
```

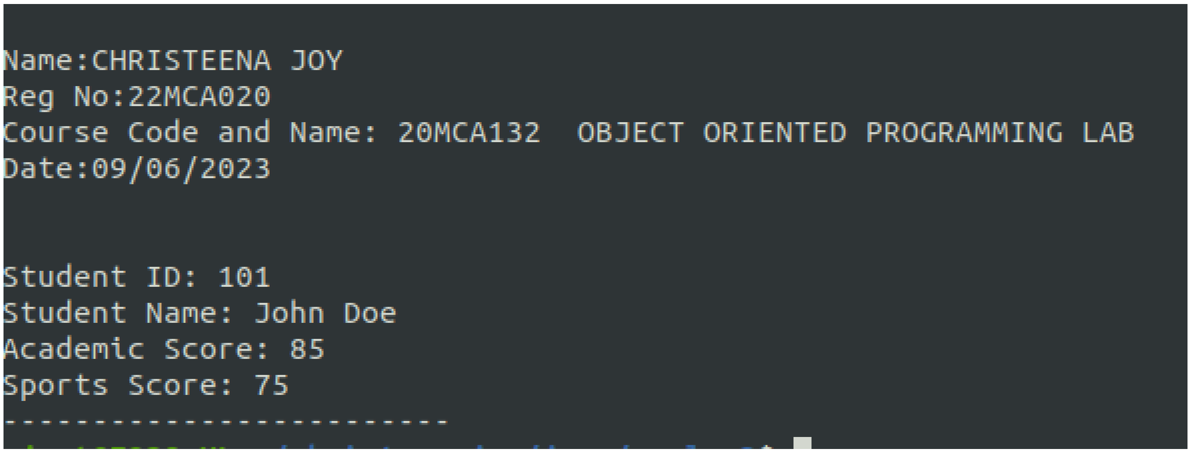
5. Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

CODE:

```
class Student {
    private int studentId;
    private String studentName;

    public Student(int studentId, String studentName) {
        this.studentId = studentId;
        this.studentName = studentName;
    }

    public int getStudentId() {
        return studentId;
    }

    public String getStudentName() {
        return studentName;
    }
}

class Sports {
    private int sportsScore;

    public Sports(int sportsScore) {
        this.sportsScore = sportsScore;
    }

    public int getSportsScore() {
        return sportsScore;
    }
}

class Result extends Student {
    private Sports sports;
    private int academicScore;

    public Result(int studentId, String studentName, int academicScore, int sportsScore) {
        super(studentId, studentName);
        this.academicScore = academicScore;
        this.sports = new Sports(sportsScore);
    }

    public int getAcademicScore() {
        return academicScore;
    }

    public int getSportsScore() {
        return sports.getSportsScore();
    }

    public void displayScores() {
        System.out.println("Student ID: " + getStudentId());
        System.out.println("Student Name: " + getStudentName());
        System.out.println("Academic Score: " + academicScore);
        System.out.println("Sports Score: " + getSportsScore());
        System.out.println("-------------------------");
    }
}

public class SportsMain {

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

```
    System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:09/06/2023\n\n");
        Result result = new Result(101, "John Doe", 85, 75);
        result.displayScores();
    }
}
```

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:09/06/2023


Student ID: 101
Student Name: John Doe
Academic Score: 85
Sports Score: 75
-------------------------
```

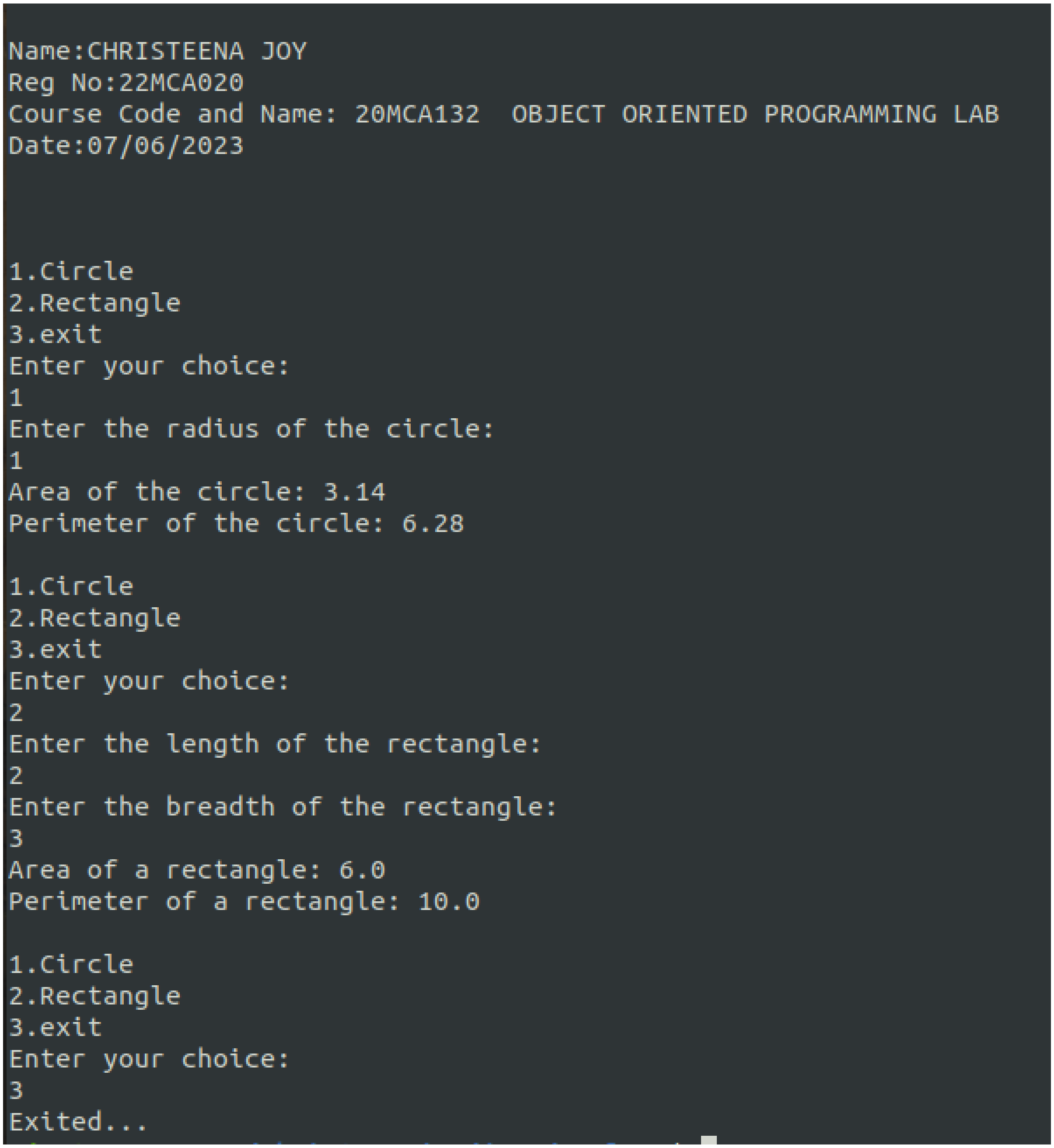
6. Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

CODE:

```
import java.util.Scanner;

interface shapes
{
    void getdata();
    void area();
    void perimeter();
}

class Circle implements shapes
{
    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("Area of the circle: "+(pi*r*r));
    }
}

class Rectangle implements shapes
{
    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("Area of a rectangle: "+(l*b));
    }
    @Override
    public void perimeter()
    {
        System.out.println("Perimeter of a rectangle: "+(2*(l+b)));
    }
}

public class ShapesMain
{
 public static void main(String[] args)
{
System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and Name:
20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:07/06/2023\n\n");
```

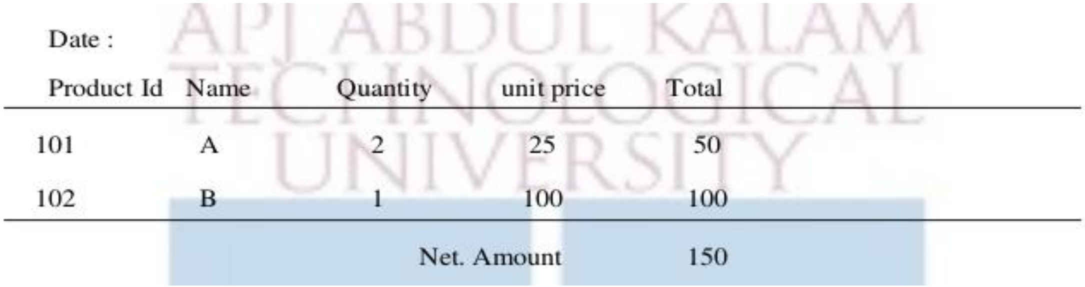
```
int ch;
Scanner sc = new Scanner(System.in);
Circle ob = new Circle();
Rectangle obj = new Rectangle();
do
{
System.out.println("\n1.Circle\n2.Rectangle\n3.exit");
System.out.println("Enter your choice:");
ch = sc.nextInt();
switch(ch)
{
case 1 :ob.getdata();
ob.area();
ob.perimeter();
break;
case 2 :obj.getdata();
obj.area();
obj.perimeter();
break;
case 3 :System.out.println("Exited...");
System.exit(0);
}
}while(true);
}
}
```

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:07/06/2023



1.Circle
2.Rectangle
3.exit
Enter your choice:
1
Enter the radius of the circle:
1
Area of the circle: 3.14
Perimeter of the circle: 6.28

1.Circle
2.Rectangle
3.exit
Enter your choice:
2
Enter the length of the rectangle:
2
Enter the breadth of the rectangle:
3
Area of a rectangle: 6.0
Perimeter of a rectangle: 10.0

1.Circle
2.Rectangle
3.exit
Enter your choice:
3
Exited...
```

7. Prepare a bill with the given format using calculate method from interface.



APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Date :

Product Id	Name	Quantity	unit price	Total
101	A	2	25	50
102	B	1	100	100
			Net. Amount	150

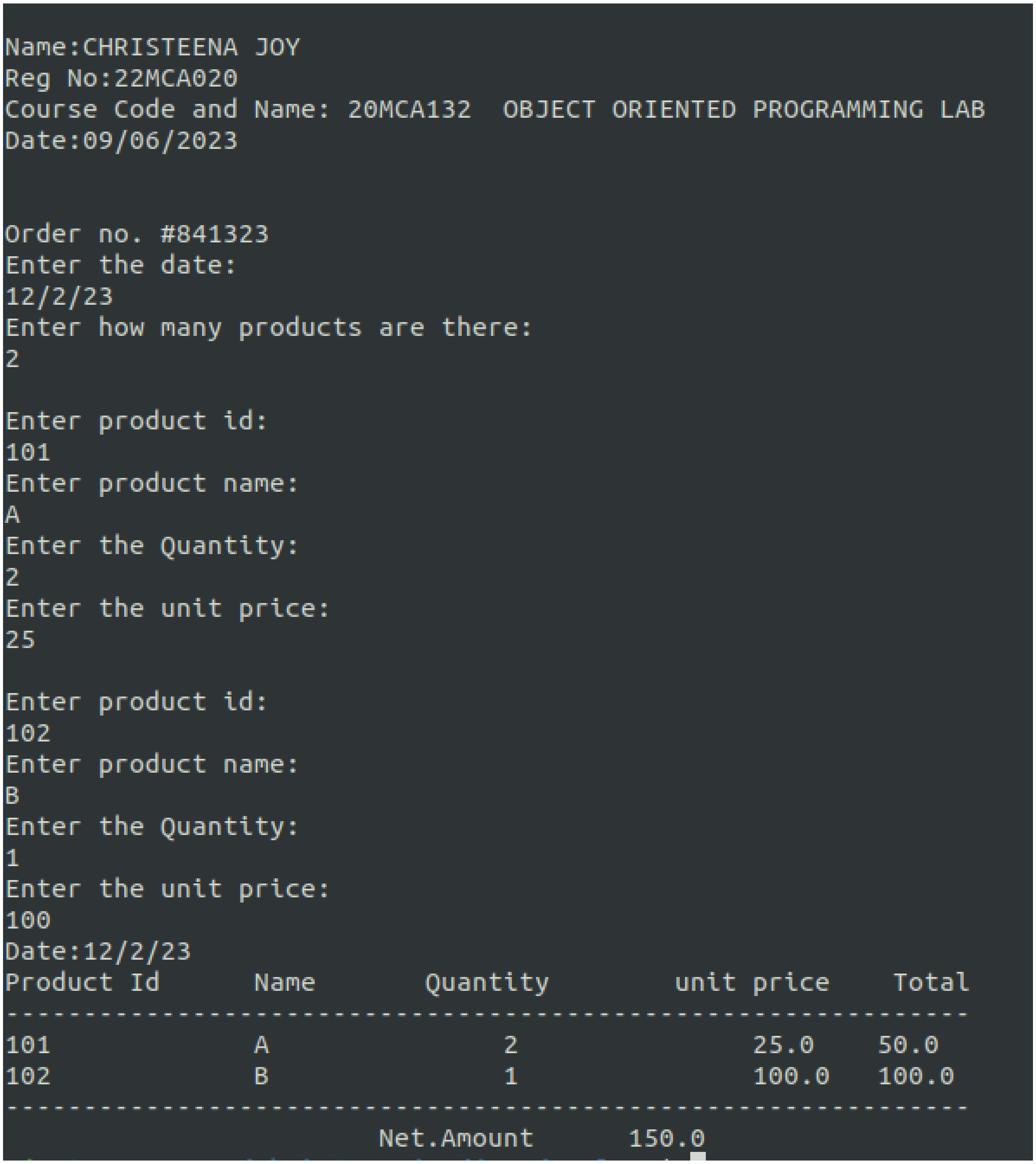
CODE:

```
import java.util.Scanner;

interface cal
{
    void calculate();
}

class bill implements cal
{
    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("\nEnter 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 Bill
{
public static void main(String[] args)
   {
         System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code
and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:09/06/2023\n\n");
      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);

  }
}
```

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:09/06/2023


Order no. #841323
Enter the date:
12/2/23
Enter how many products are there:
2

Enter product id:
101
Enter product name:
A
Enter the Quantity:
2
Enter the unit price:
25

Enter product id:
102
Enter product name:
B
Enter the Quantity:
1
Enter the unit price:
100
Date:12/2/23
Product Id      Name       Quantity        unit price    Total
--------------------------------------------------------------
101             A               2               25.0    50.0
102             B               1               100.0   100.0
--------------------------------------------------------------
                        Net.Amount      150.0
```

Cycle 4

1. 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.

CODE:

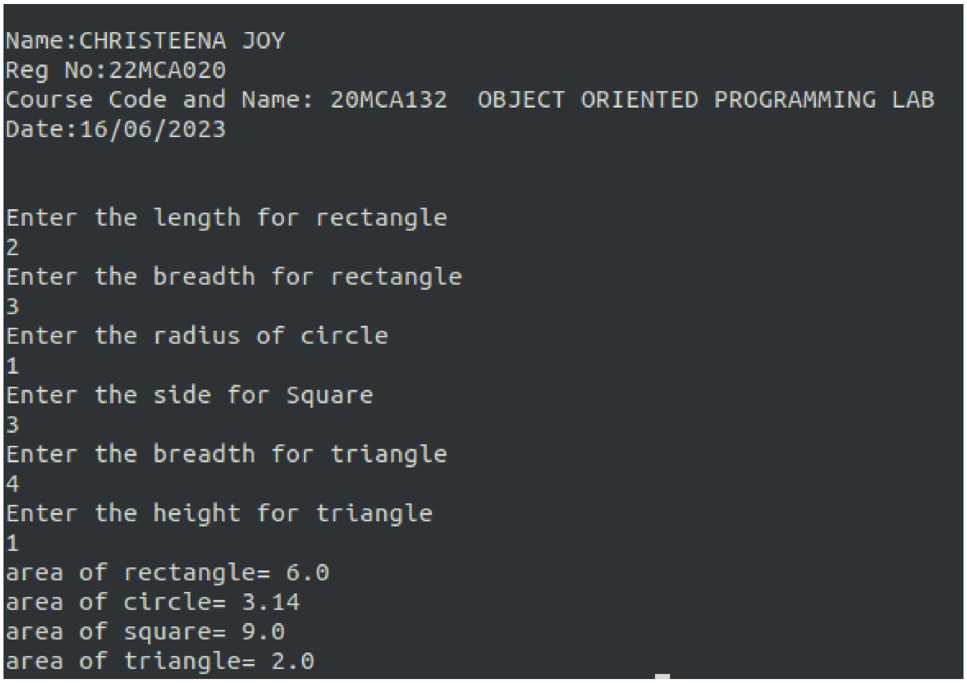
package_graphics.java

```
package package_graphics;
interface interface_graphics{
public float recArea(int l, int h);
public float cirArea(int r);
public float squArea(int a);
public float triArea(int l, int h);
}

public class package_graphics implements interface_graphics {

public float recArea(int l, int h){
return l*h;
}
public float cirArea(int r){
return r*r*(float)3.14;
}
public float squArea(int a){
return a*a;
}
public float triArea(int l, int h){
return l*h*(float)(.5);
}
}
```

main_graphics.java

```
import package_graphics.*;
import java.util.*;
public class main_graphics {

public static void main(String []args){
System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and Name:
20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:16/06/2023\n\n");
package_graphics testObj = new package_graphics();
int l,h,r,a,c,d;
Scanner s=new Scanner(System.in);
System.out.println("Enter the length for rectangle");
l=s.nextInt();
System.out.println("Enter the breadth for rectangle");
h=s.nextInt();
System.out.println("Enter the radius of circle");
r=s.nextInt();
System.out.println("Enter the side for Square");
a=s.nextInt();
System.out.println("Enter the breadth for triangle");
c=s.nextInt();
System.out.println("Enter the height for triangle");
d=s.nextInt();
System.out.println("area of rectangle= "+testObj.recArea(l,h));
System.out.println("area of circle= "+testObj.cirArea(r));
System.out.println("area of square= "+testObj.squArea(a));
System.out.println("area of triangle= "+testObj.triArea(c,d));
}
}
```

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:16/06/2023


Enter the length for rectangle
2
Enter the breadth for rectangle
3
Enter the radius of circle
1
Enter the side for Square
3
Enter the breadth for triangle
4
Enter the height for triangle
1
area of rectangle= 6.0
area of circle= 3.14
area of square= 9.0
area of triangle= 2.0
```

2. 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.

CODE:

Arithmetic package:

Addition.java

```
package arithmetic;

public interface Addition {
    public double add(double num1, double num2);
}
```

Subtraction.java

```
package arithmetic;

public interface Subtraction {
    public double subtract(double num1, double num2);
}
```

Multiplication.java

```
package arithmetic;

public interface Multiplication {
    public double multiply(double num1, double num2);
}
```

Division.java

```
package arithmetic;

public interface Division {
    public double divide(double num1, double num2);
}
```

ArithmeticOperations.java

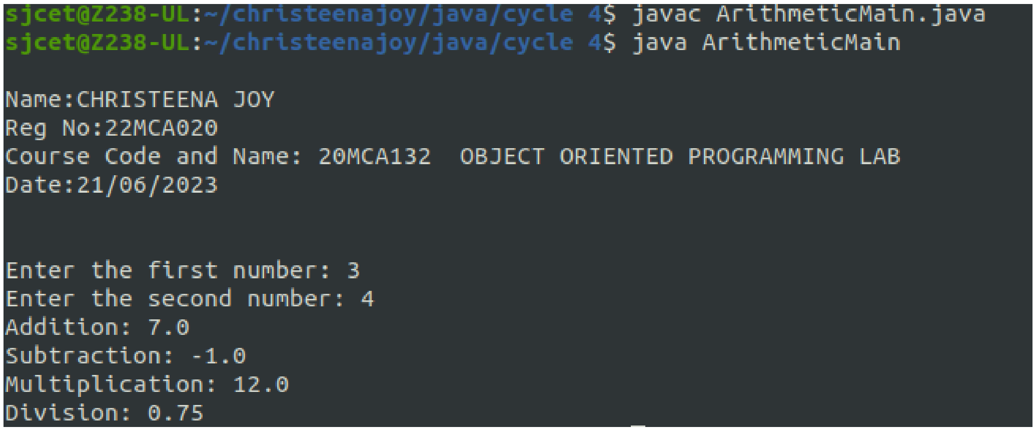
```
package arithmetic;

public class ArithmeticOperations implements Addition, Subtraction, Multiplication, Division {
    @Override
    public double add(double num1, double num2) {
        return num1 + num2;
    }

    @Override
    public double subtract(double num1, double num2) {
        return num1 - num2;
    }

    @Override
    public double multiply(double num1, double num2) {
        return num1 * num2;
    }

    @Override
    public double divide(double num1, double num2) {
        if (num2 == 0) {
            throw new ArithmeticException("Division by zero error!");
        }
        return num1 / num2;
    }
}
```

ArithmeticMain.java

```
import arithmetic.ArithmeticOperations;
import java.util.Scanner;
```

```
public class ArithmeticMain {
   public static void main(String[] args) {
   System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and 
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:21/06/2023\n\n");
      ArithmeticOperations operations = new ArithmeticOperations();

     Scanner scanner = new Scanner(System.in);

      System.out.print("Enter the first number: ");
      double num1 = scanner.nextDouble();

      System.out.print("Enter the second number: ");
      double num2 = scanner.nextDouble();

      System.out.println("Addition: " + operations.add(num1, num2));
      System.out.println("Subtraction: " + operations.subtract(num1, num2));
      System.out.println("Multiplication: " + operations.multiply(num1, num2));
      System.out.println("Division: " + operations.divide(num1, num2));

   }
}
```

OUTPUT:



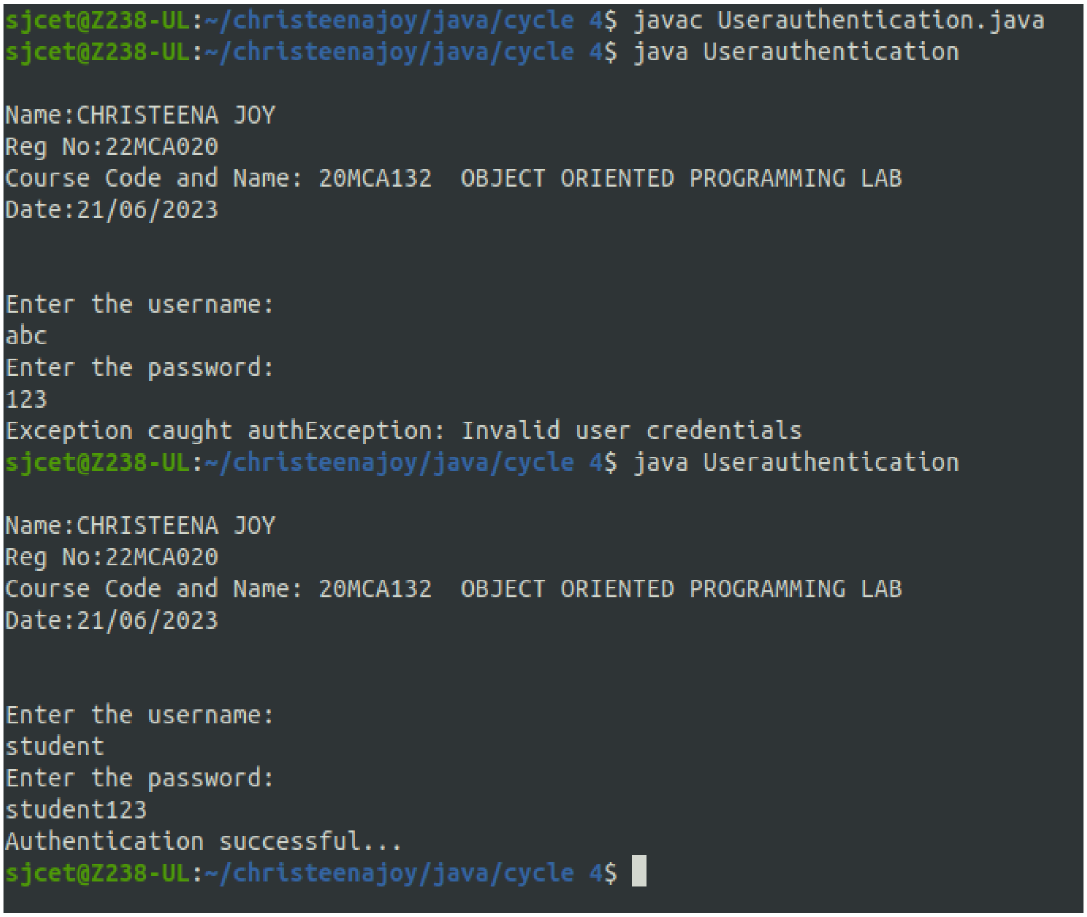
```
sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ javac ArithmeticMain.java
sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ java ArithmeticMain

Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:21/06/2023


Enter the first number: 3
Enter the second number: 4
Addition: 7.0
Subtraction: -1.0
Multiplication: 12.0
Division: 0.75
```

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

CODE:

```
import java.util.Scanner;
class authException extends Exception
{
public authException(String s) {
super(s);
}
}
public class Userauthentication
{
public static void main(String[] args) {
System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and Name:
20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:21/06/2023\n\n");
String username = "student";
String passcode = "student123";
String user_name,password;
Scanner sc = new Scanner(System.in);
try
{
System.out.println("Enter the username:");
user_name = sc.nextLine();
System.out.println("Enter the password:");
password = sc.nextLine();
if(username.equals(user_name) && passcode.equals(password))
{
System.out.println("Authentication successful...");
}
else
throw new authException("Invalid user credentials");
}
catch(authException e)
{
System.out.println("Exception caught "+e);
}
}
}
```

OUTPUT:



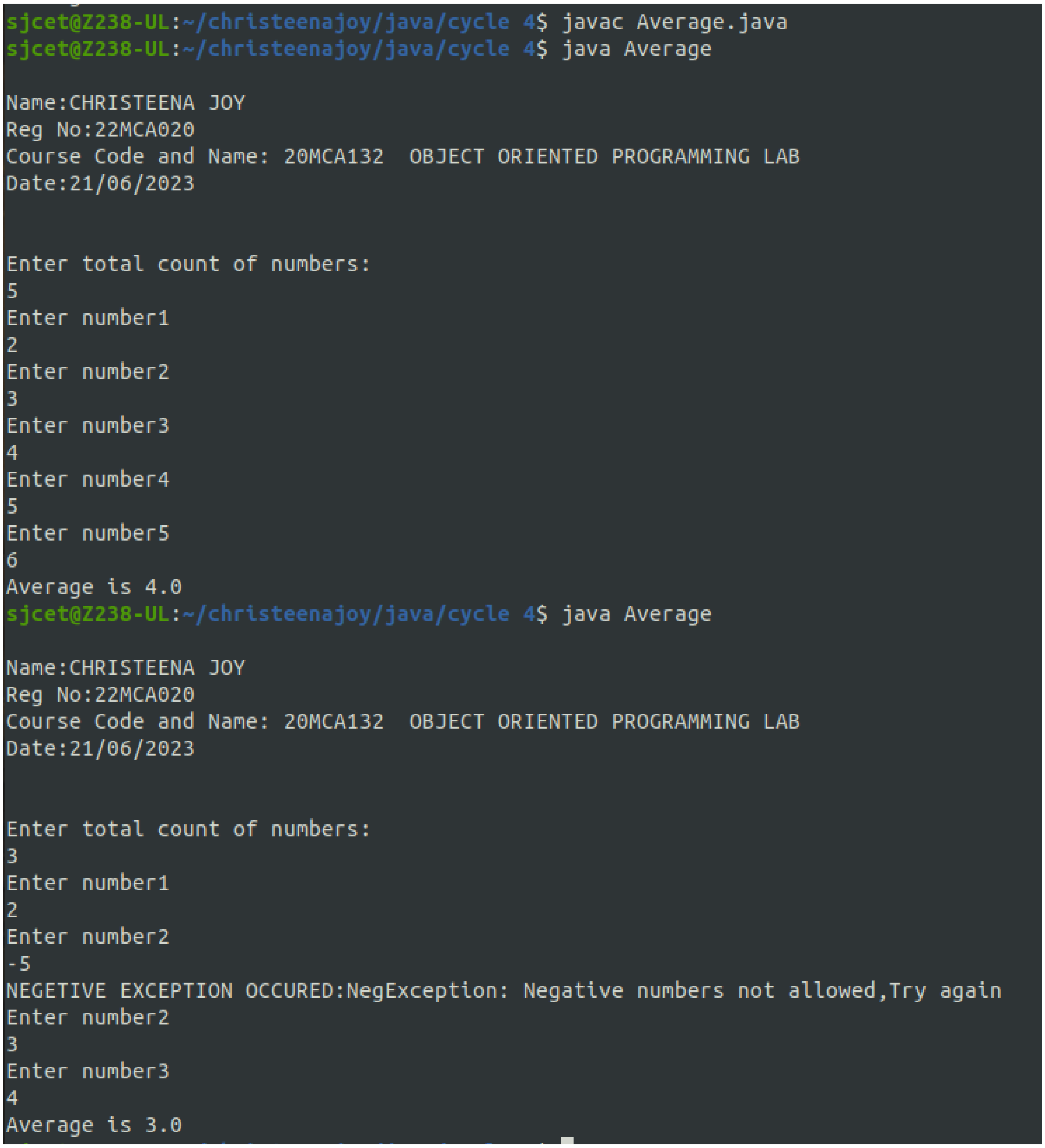
```
sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ javac Userauthentication.java
sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ java Userauthentication

Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:21/06/2023


Enter the username:
abc
Enter the password:
123
Exception caught authException: Invalid user credentials
sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ java Userauthentication

Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:21/06/2023


Enter the username:
student
Enter the password:
student123
Authentication successful...
sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ 
```

4. Find the average of N positive integers, raising a user defined exception for each negative Input.

CODE:

```
import java.util.Scanner;
class NegException extends Exception
{
public NegException(String s)
{
super(s);
}
}

public class Average {
public static void main(String[] args)
{
System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and Name:
20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:21/06/2023\n\n");
int i;
double sum=0,avg=0;
Scanner sc=new Scanner(System.in);
System.out.println("Enter total count of numbers:");
int n=sc.nextInt();

for(i=1;i<=n;i++)
{
try
{
System.out.println("Enter number"+i);
int a=sc.nextInt();
if(a<0)
{
i--;
throw new NegException("Negative numbers not allowed,Try again");
}
else
{
sum=sum+a;
}
}
catch(NegException e)
{
System.out.println("NEGETIVE EXCEPTION OCCURED:"+e);
}
}
avg=sum/n;
System.out.println("Average is "+avg);
sc.close();
}

}
```

OUTPUT:



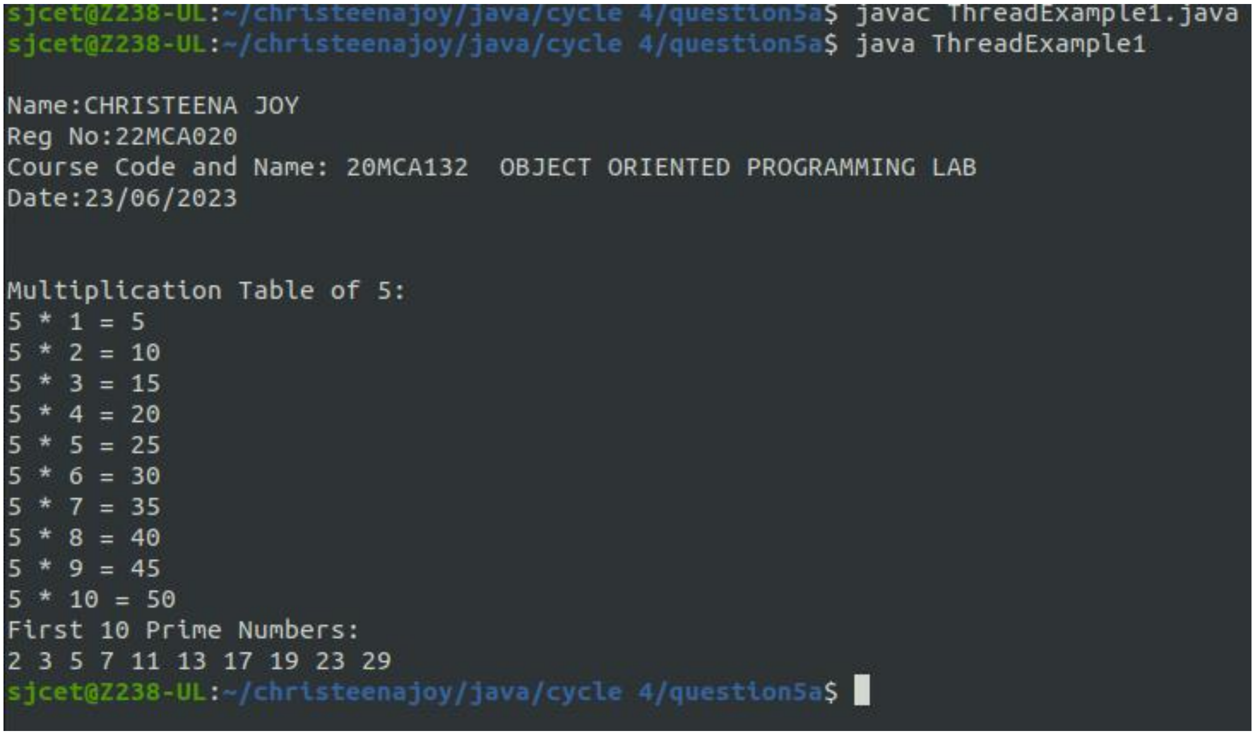
```
sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ javac Average.java
sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ java Average

Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:21/06/2023


Enter total count of numbers:
5
Enter number1
2
Enter number2
3
Enter number3
4
Enter number4
5
Enter number5
6
Average is 4.0
sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ java Average

Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:21/06/2023


Enter total count of numbers:
3
Enter number1
2
Enter number2
-5
NEGETIVE EXCEPTION OCCURED:NegException: Negative numbers not allowed,Try again
Enter number2
3
Enter number3
4
Average is 3.0
```

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

CODE:

```
class MultiplicationTable implements Runnable {
    @Override
    public void run() {
        System.out.println("Multiplication Table of 5:");
        for (int i = 1; i <= 10; i++) {
            System.out.println("5 * " + i + " = " + (5 * i));
        }
    }
}

class PrimeNumbers implements Runnable {
    @Override
    public void run() {
        System.out.println("First 10 Prime Numbers:");
        int count = 0;
        int num = 2;
        while (count < 10) {
            if (isPrime(num)) {
                System.out.print(num + " ");
                count++;
            }
            num++;
        }
        System.out.println();
    }

    private boolean isPrime(int num) {
        if (num < 2) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }
}

public class ThreadExample1 {
    public static void main(String[] args) {
    System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and 
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:23/06/2023\n\n");
        MultiplicationTable multiplicationTable = new MultiplicationTable();
        PrimeNumbers primeNumbers = new PrimeNumbers();

        Thread thread1 = new Thread(multiplicationTable);
        Thread thread2 = new Thread(primeNumbers);

        thread1.start();
        try {
            thread1.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        thread2.start();
    }
}
```

OUTPUT:



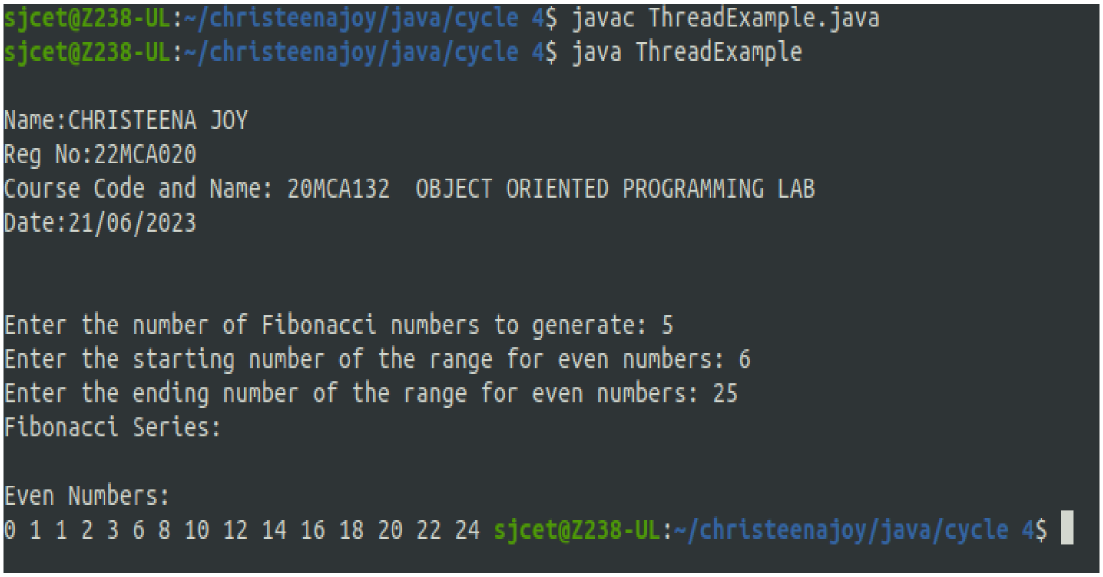
```
sjcet@Z238-UL:~/christeenajoy/java/cycle 4/question5a$ javac ThreadExample1.java
sjcet@Z238-UL:~/christeenajoy/java/cycle 4/question5a$ java ThreadExample1

Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:23/06/2023


Multiplication Table of 5:
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
First 10 Prime Numbers:
2 3 5 7 11 13 17 19 23 29
sjcet@Z238-UL:~/christeenajoy/java/cycle 4/question5a$ 
```

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

CODE:

```
import java.util.Scanner;
class FibonacciGenerator implements Runnable {
    private int n;

    public FibonacciGenerator(int n) {
        this.n = n;
    }

    public void run() {
        int prev = 0;
        int curr = 1;

        System.out.println("Fibonacci Series:");
        System.out.print(prev + " ");

        for (int i = 1; i < n; i++) {
            System.out.print(curr + " ");

            int temp = prev + curr;
            prev = curr;
            curr = temp;
        }
    }
}

class EvenNumberPrinter implements Runnable {
    private int start;
    private int end;
    public EvenNumberPrinter(int start, int end) {
        this.start = start;
        this.end = end;
    }

    public void run() {
        System.out.println("\nEven Numbers:");
        for (int i = start; i <= end; i++) {
            if (i % 2 == 0) {
                System.out.print(i + " ");
            }
        }
    }
}

public class ThreadExample {
    public static void main(String[] args) {
    System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and 
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:21/06/2023\n\n");
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of Fibonacci numbers to generate: ");
        int n = scanner.nextInt();

        System.out.print("Enter the starting number of the range for even numbers: ");
        int start = scanner.nextInt();

        System.out.print("Enter the ending number of the range for even numbers: ");
        int end = scanner.nextInt();

        // Create threads for Fibonacci generation and even number printing
        Thread fibonacciThread = new Thread(new FibonacciGenerator(n));
        Thread evenNumberThread = new Thread(new EvenNumberPrinter(start, end));
        // Start the threads
        fibonacciThread.start();
        evenNumberThread.start();
    }
}
```

OUTPUT:



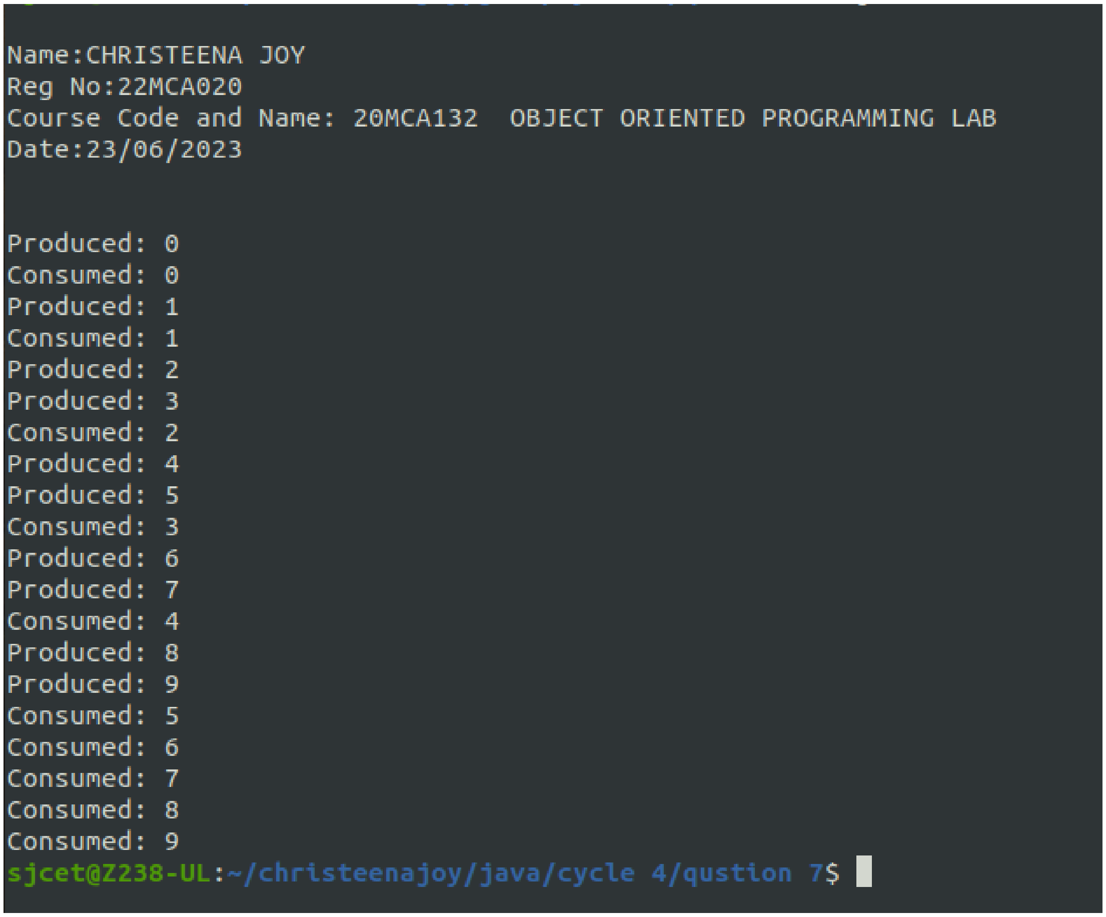
```
sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ javac ThreadExample.java
sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ java ThreadExample

Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:21/06/2023


Enter the number of Fibonacci numbers to generate: 5
Enter the starting number of the range for even numbers: 6
Enter the ending number of the range for even numbers: 25
Fibonacci Series:

Even Numbers:
0 1 1 2 3 6 8 10 12 14 16 18 20 22 24 sjcet@Z238-UL:~/christeenajoy/java/cycle 4$ 
```

7. Producer/Consumer using ITC

CODE:

```
import java.util.LinkedList;

class Buffer {
    private LinkedList<Integer> buffer;
    private int capacity;

    public Buffer(int capacity) {
        this.buffer = new LinkedList<>();
        this.capacity = capacity;
    }

    public void produce(int value) throws InterruptedException {
        synchronized (this) {
            while (buffer.size() == capacity) {
                wait();
            }

            buffer.add(value);
            System.out.println("Produced: " + value);
            notifyAll();
        }
    }

    public void consume() throws InterruptedException {
        synchronized (this) {
            while (buffer.isEmpty()) {
                wait();
            }

            int value = buffer.removeFirst();
            System.out.println("Consumed: " + value);
            notifyAll();
        }
    }
}

class Producer implements Runnable {
    private Buffer buffer;
    private int numProductions;

    public Producer(Buffer buffer, int numProductions) {
        this.buffer = buffer;
        this.numProductions = numProductions;
    }

    @Override
    public void run() {
        for (int i = 0; i < numProductions; i++) {
            try {
                buffer.produce(i);
                Thread.sleep(1000); // Simulate production time
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class Consumer implements Runnable {
    private Buffer buffer;
    private int numConsumptions;

    public Consumer(Buffer buffer, int numConsumptions) {
        this.buffer = buffer;
        this.numConsumptions = numConsumptions;
    }

    @Override
    public void run() {
        for (int i = 0; i < numConsumptions; i++) {
            try {
                buffer.consume();
                Thread.sleep(2000); // Simulate consumption time
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class ProducerConsumerITC {
    public static void main(String[] args) {
    System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:23/06/2023\n\n");
        Buffer buffer = new Buffer(5);
        int numProductions = 10;
        int numConsumptions = 10;

        Producer producer = new Producer(buffer, numProductions);
        Consumer consumer = new Consumer(buffer, numConsumptions);

        Thread producerThread = new Thread(producer);
        Thread consumerThread = new Thread(consumer);

        producerThread.start();
        consumerThread.start();
    }
}
```

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:23/06/2023


Produced: 0
Consumed: 0
Produced: 1
Consumed: 1
Produced: 2
Produced: 3
Consumed: 2
Produced: 4
Produced: 5
Consumed: 3
Produced: 6
Produced: 7
Consumed: 4
Produced: 8
Produced: 9
Consumed: 5
Consumed: 6
Consumed: 7
Consumed: 8
Consumed: 9
sjcet@Z238-UL:~/christeenajoy/java/cycle 4/qustion 7$
```

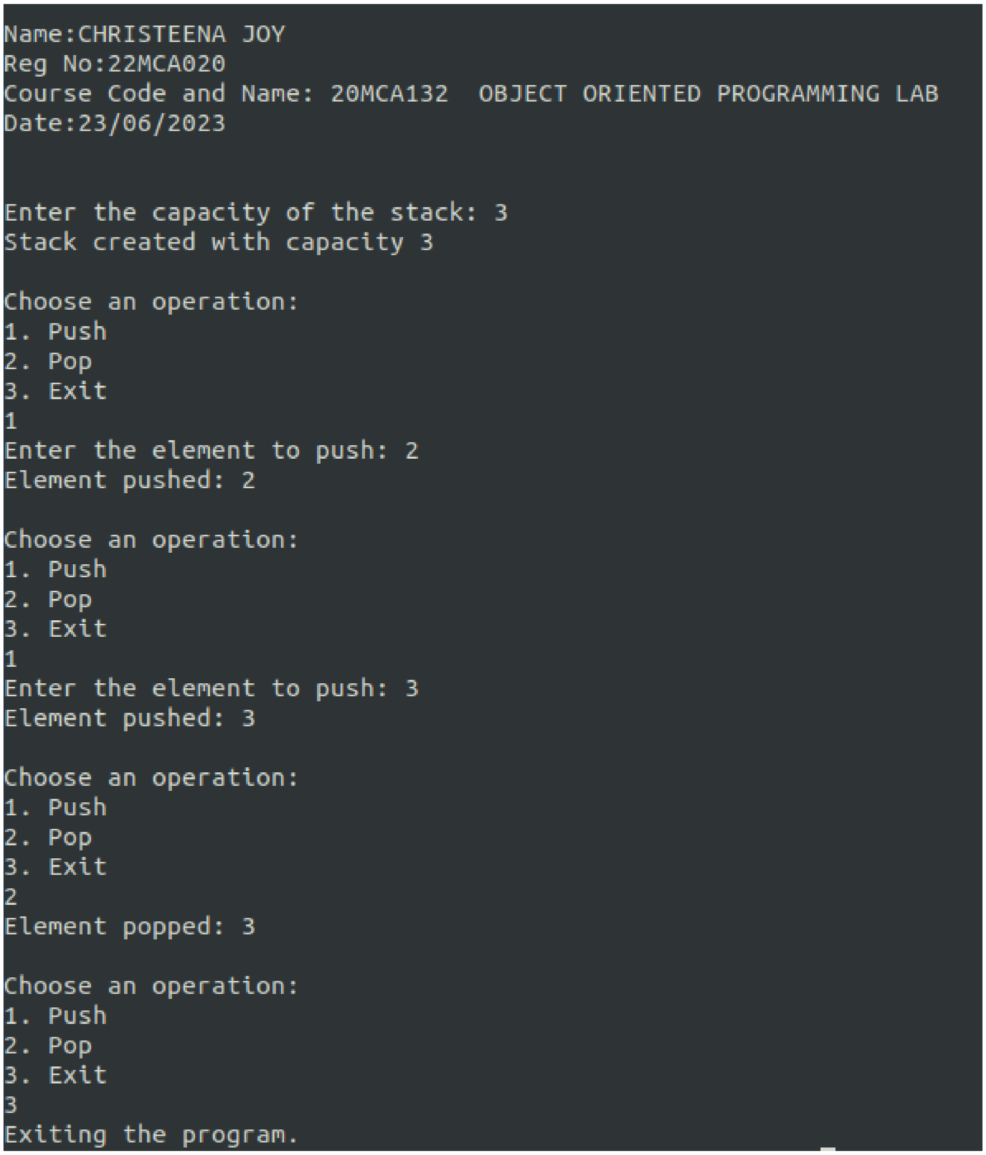
8. Program to create a generic stack and do the Push and Pop operations.

CODE:

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Stack<T> {
    private List<T> stack;
    private int capacity;

    public Stack(int capacity) {
        this.stack = new ArrayList<>();
        this.capacity = capacity;
    }

    public void push(T element) {
        if (stack.size() >= capacity) {
            throw new IllegalStateException("Stack is full");
        }
        stack.add(element);
    }

    public T pop() {
        if (isEmpty()) {
            throw new IllegalStateException("Stack is empty");
        }
        return stack.remove(stack.size() - 1);
    }

    public boolean isEmpty() {
        return stack.isEmpty();
    }
```

```
}

public class GenericStackExample {
    public static void main(String[] args) {
    System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and 
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:23/06/2023\n\n");
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the capacity of the stack: ");
        int capacity = scanner.nextInt();

        Stack<Integer> integerStack = new Stack<>(capacity);

        System.out.println("Stack created with capacity " + capacity);

        while (true) {
            System.out.println("\nChoose an operation:");
            System.out.println("1. Push");
            System.out.println("2. Pop");
            System.out.println("3. Exit");

            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter the element to push: ");
                    int element = scanner.nextInt();
                    try {
                        integerStack.push(element);
                        System.out.println("Element pushed: " + element);
                    } catch (IllegalStateException e) {
                        System.out.println("Stack is full. Cannot push element.");
                    }
                    break;
                case 2:
                    try {
                        int poppedElement = integerStack.pop();
                        System.out.println("Element popped: " + poppedElement);
                    } catch (IllegalStateException e) {
                        System.out.println("Stack is empty. Cannot pop element.");
                    }
                    break;
                case 3:
                    System.out.println("Exiting the program.");
                    scanner.close();
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please try again.");
                    break;
            }
        }
    }
}
```

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:23/06/2023


Enter the capacity of the stack: 3
Stack created with capacity 3

Choose an operation:
1. Push
2. Pop
3. Exit
1
Enter the element to push: 2
Element pushed: 2

Choose an operation:
1. Push
2. Pop
3. Exit
1
Enter the element to push: 3
Element pushed: 3

Choose an operation:
1. Push
2. Pop
3. Exit
2
Element popped: 3

Choose an operation:
1. Push
2. Pop
3. Exit
3
Exiting the program.
```

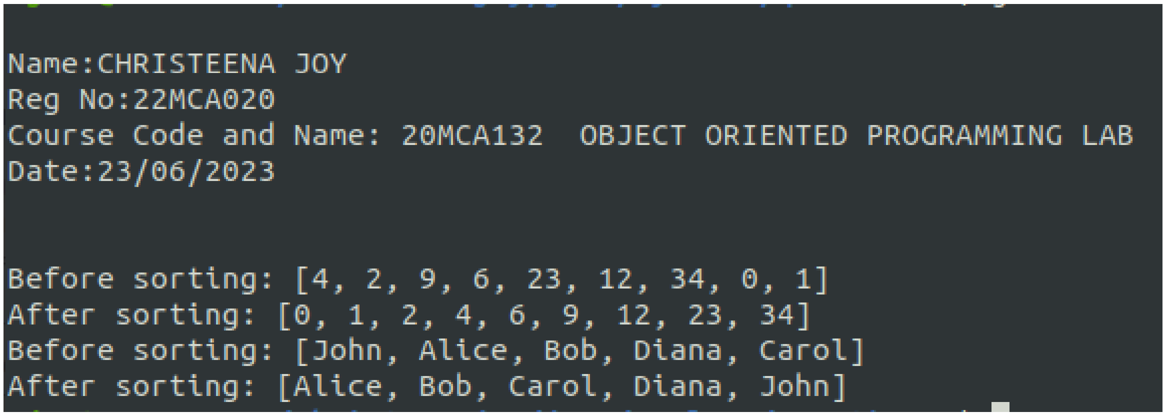
9. Using generic method perform Bubble sort.

CODE:

```
import java.util.Arrays;

public class BubbleSort1 {
    public static <T extends Comparable<T>> void bubbleSort(T[] arr) {
        int n = arr.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j].compareTo(arr[j + 1]) > 0) {
                    // Swap arr[j] and arr[j + 1]
                    T temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }

    public static void main(String[] args) {
    System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:23/06/2023\n\n");
        Integer[] numbers = { 4, 2, 9, 6, 23, 12, 34, 0, 1 };
        String[] names = { "John", "Alice", "Bob", "Diana", "Carol" };

        System.out.println("Before sorting: " + Arrays.toString(numbers));
        bubbleSort(numbers);
        System.out.println("After sorting: " + Arrays.toString(numbers));

        System.out.println("Before sorting: " + Arrays.toString(names));
        bubbleSort(names);
        System.out.println("After sorting: " + Arrays.toString(names));
```

```
  }
}
```

OUTPUT:



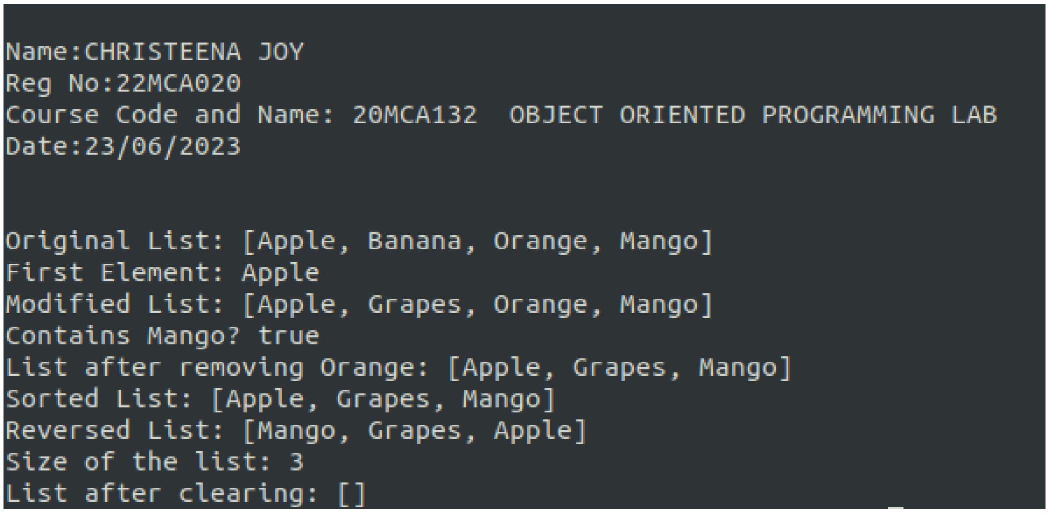
```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:23/06/2023


Before sorting: [4, 2, 9, 6, 23, 12, 34, 0, 1]
After sorting: [0, 1, 2, 4, 6, 9, 12, 23, 34]
Before sorting: [John, Alice, Bob, Diana, Carol]
After sorting: [Alice, Bob, Carol, Diana, John]
```

10. Maintain a list of Strings using ArrayList from collection framework, perform built-in Operations.

CODE:

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class ArrayListExample {
    public static void main(String[] args) {
    System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:23/06/2023\n\n");
        // Create an ArrayList to store strings
        List<String> stringList = new ArrayList<>();

        // Add elements to the list
        stringList.add("Apple");
        stringList.add("Banana");
        stringList.add("Orange");
        stringList.add("Mango");

        // Print the original list
        System.out.println("Original List: " + stringList);

        // Accessing elements by index
        String firstElement = stringList.get(0);
        System.out.println("First Element: " + firstElement);

        // Modifying an element
        stringList.set(1, "Grapes");
        System.out.println("Modified List: " + stringList);

        // Checking if an element exists in the list
        boolean containsMango = stringList.contains("Mango");
        System.out.println("Contains Mango? " + containsMango);

        // Removing an element
        stringList.remove("Orange");
        System.out.println("List after removing Orange: " + stringList);

        // Sorting the list
        Collections.sort(stringList);
        System.out.println("Sorted List: " + stringList);

        // Reversing the list
        Collections.reverse(stringList);
        System.out.println("Reversed List: " + stringList);

        // Size of the list
        int size = stringList.size();
        System.out.println("Size of the list: " + size);

        // Clearing the list
        stringList.clear();
        System.out.println("List after clearing: " + stringList);
    }
}
```

OUTPUT:



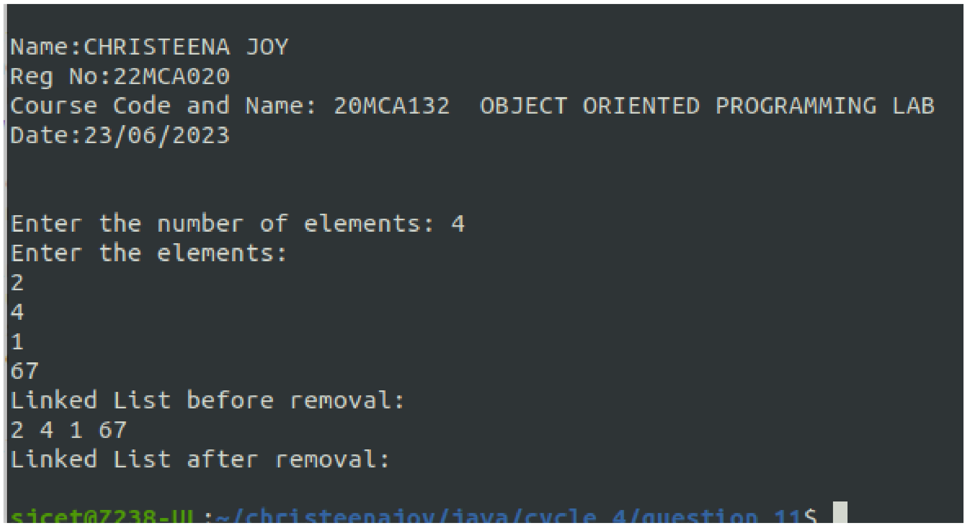
```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:23/06/2023


Original List: [Apple, Banana, Orange, Mango]
First Element: Apple
Modified List: [Apple, Grapes, Orange, Mango]
Contains Mango? true
List after removing Orange: [Apple, Grapes, Mango]
Sorted List: [Apple, Grapes, Mango]
Reversed List: [Mango, Grapes, Apple]
Size of the list: 3
List after clearing: []
```

11. Program to remove all the elements from a linked list.

CODE:

```
import java.util.Scanner;

public class LinkedListDemo {
    private Node head;

    private class Node {
        String data;
        Node next;

        Node(String data) {
            this.data = data;
            this.next = null;
        }
    }

    public void add(String data) {
        Node newNode = new Node(data);

        if (head == null) {
            head = newNode;
        } else {
            Node currentNode = head;
            while (currentNode.next != null) {
                currentNode = currentNode.next;
            }
            currentNode.next = newNode;
        }
    }

    public void removeAll() {
        head = null;
    }

    public void display() {
        Node currentNode = head;
        while (currentNode != null) {
            System.out.print(currentNode.data + " ");
            currentNode = currentNode.next;
        }
        System.out.println();
    }

    public static void main(String[] args) {
    System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and 
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:23/06/2023\n\n");
        LinkedListDemo linkedList = new LinkedListDemo();
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements: ");
        int numElements = scanner.nextInt();

        System.out.println("Enter the elements:");
        for (int i = 0; i < numElements; i++) {
            String element = scanner.next();
            linkedList.add(element);
        }

        System.out.println("Linked List before removal:");
        linkedList.display();

        linkedList.removeAll();

        System.out.println("Linked List after removal:");
        linkedList.display();
        scanner.close();
    }
}
```

OUTPUT:



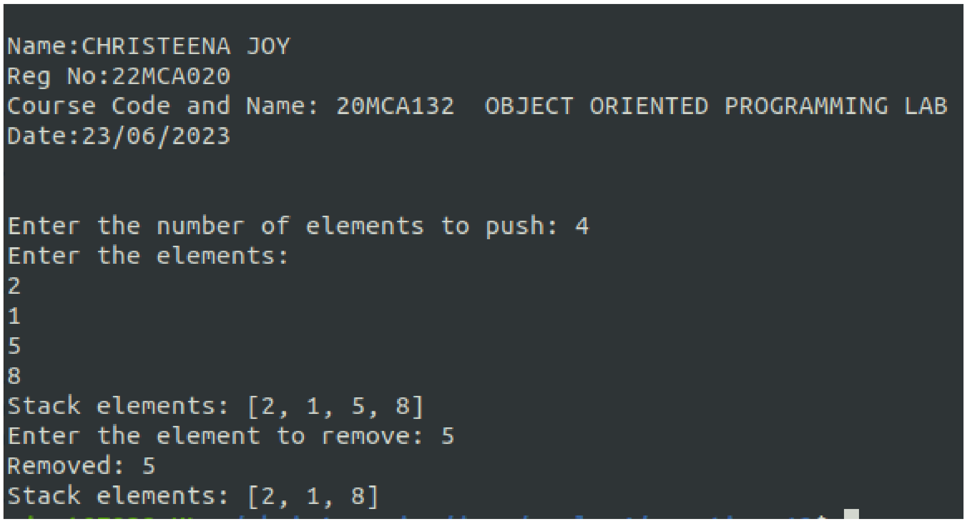
```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:23/06/2023


Enter the number of elements: 4
Enter the elements:
2
4
1
67
Linked List before removal:
2 4 1 67
Linked List after removal:

sjcet@Z238-UL:~/christeenajoy/java/cycle 4/question 11$
```

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

CODE:

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Stack {
    private List<Object> elements;

    public Stack() {
        this.elements = new ArrayList<>();
    }

    public void push(Object item) {
        elements.add(item);
    }

    public Object pop() {
        if (elements.isEmpty()) {
            System.out.println("Stack is empty.");
            return null;
        }

        return elements.remove(elements.size() - 1);
    }

    public void remove(Object item) {
        if (elements.contains(item)) {
            elements.remove(item);
            System.out.println("Removed: " + item);
        } else {
            System.out.println("Item not found: " + item);
        }
    }

    public void display() {
        System.out.println("Stack elements: " + elements);
    }
}

public class StackMain {

    public static void main(String[] args) {
        System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:23/06/2023\n\n");
        Stack stack = new Stack();
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements to push: ");
        int numElements = scanner.nextInt();

        System.out.println("Enter the elements:");
        for (int i = 0; i < numElements; i++) {
            Object element = scanner.next();
            stack.push(element);
        }

        stack.display();

        System.out.print("Enter the element to remove: ");
        Object elementToRemove = scanner.next();
        stack.remove(elementToRemove);

        stack.display();
    }
}
```

OUTPUT:



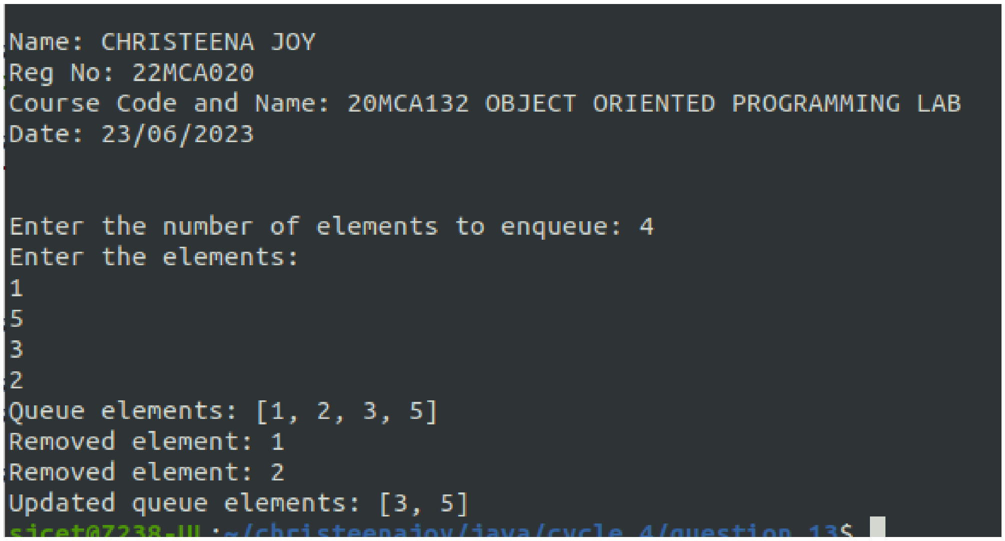
```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:23/06/2023


Enter the number of elements to push: 4
Enter the elements:
2
1
5
8
Stack elements: [2, 1, 5, 8]
Enter the element to remove: 5
Removed: 5
Stack elements: [2, 1, 8]
```

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

<u>CODE:</u>

```
import java.util.PriorityQueue;
import java.util.Queue;
import java.util.Scanner;

public class PriorityQueueDemo {
    public static void main(String[] args) {
        System.out.println("\nName: CHRISTEENA JOY\nReg No: 22MCA020\nCourse Code and Name: 20MCA132 OBJECT ORIENTED PROGRAMMING LAB\nDate: 23/06/2023\n\n");

        Scanner scanner = new Scanner(System.in);
        Queue<Integer> queue = new PriorityQueue<>();

        System.out.print("Enter the number of elements to enqueue: ");
        int numElements = scanner.nextInt();

        System.out.println("Enter the elements:");
        for (int i = 0; i < numElements; i++) {
            int element = scanner.nextInt();
            queue.offer(element);
        }

        System.out.println("Queue elements: " + queue);

        System.out.println("Removed element: " + queue.poll());
        System.out.println("Removed element: " + queue.poll());

        System.out.println("Updated queue elements: " + queue);
    }
}
```

OUTPUT:



```
Name: CHRISTEENA JOY
Reg No: 22MCA020
Course Code and Name: 20MCA132 OBJECT ORIENTED PROGRAMMING LAB
Date: 23/06/2023


Enter the number of elements to enqueue: 4
Enter the elements:
1
5
3
2
Queue elements: [1, 2, 3, 5]
Removed element: 1
Removed element: 2
Updated queue elements: [3, 5]
sjcet@Z238-UU:~/christeenajoy/java/cycle 4/question 13$
```

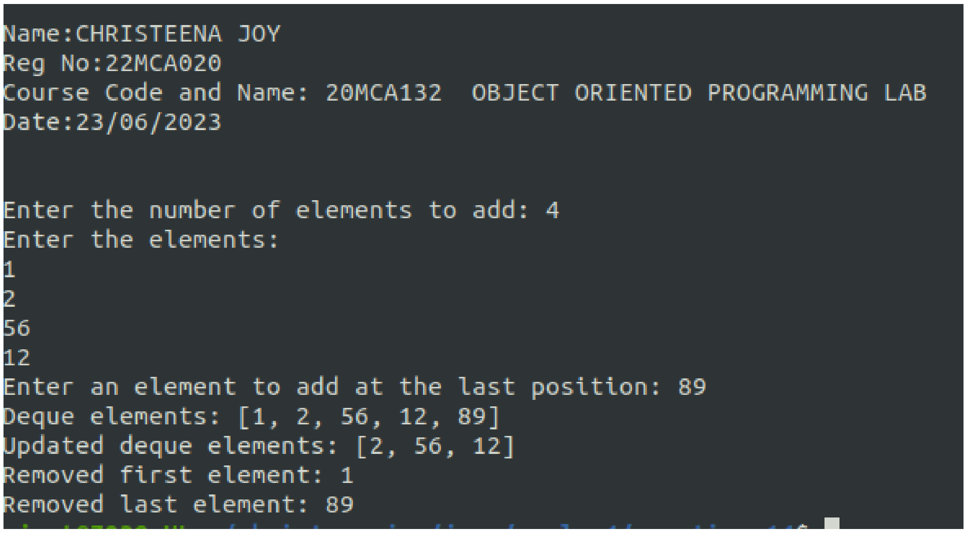
14. Program to demonstrate the addition and deletion of elements in deque

CODE:

```
import java.util.Deque;
import java.util.LinkedList;
import java.util.Scanner;

public class DequeDemo {
   public static void main(String[] args) {
   System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:23/06/2023\n\n");
      // Create a Deque
      Deque<Integer> deque = new LinkedList<>();

      // Create a Scanner for user input
      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter the number of elements to add: ");
      int numElements = scanner.nextInt();

      System.out.println("Enter the elements:");

      // Add elements at the end
      for (int i = 0; i < numElements; i++) {
         int element = scanner.nextInt();
         deque.addLast(element);
      }

      System.out.print("Enter an element to add at the last position: ");
      int lastElement = scanner.nextInt();
      deque.addLast(lastElement);

      // Print the deque
```

```
        System.out.println("Deque elements: " + deque);

        // Remove elements from the beginning and end
        int firstElement = deque.removeFirst();
        int removedLastElement = deque.removeLast();

        // Print the updated deque
        System.out.println("Updated deque elements: " + deque);
        System.out.println("Removed first element: " + firstElement);
        System.out.println("Removed last element: " + removedLastElement);

        // Close the scanner
        scanner.close();
    }
}
```

OUTPUT:



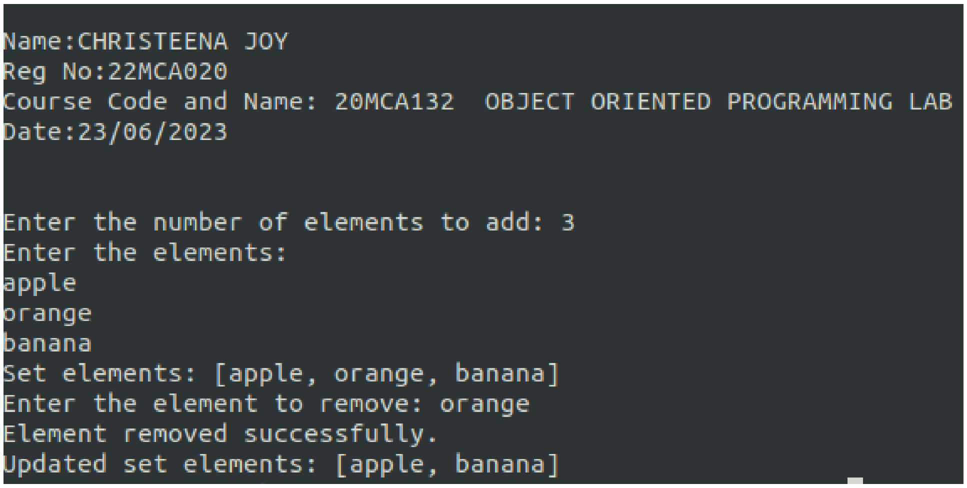
```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:23/06/2023


Enter the number of elements to add: 4
Enter the elements:
1
2
56
12
Enter an element to add at the last position: 89
Deque elements: [1, 2, 56, 12, 89]
Updated deque elements: [2, 56, 12]
Removed first element: 1
Removed last element: 89
```

15. Program to demonstrate the creation of Set object using the LinkedHashset class

CODE:

```
import java.util.LinkedHashSet;
import java.util.Scanner;
import java.util.Set;

public class SetDemo {
    public static void main(String[] args) {
        System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:23/06/2023\n\n");
        // Create a Set using LinkedHashSet
        Set<Object> set = new LinkedHashSet<>();

        // Create a Scanner for user input
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements to add: ");
        int numElements = scanner.nextInt();

        System.out.println("Enter the elements:");

        // Add elements to the set
        for (int i = 0; i < numElements; i++) {
            Object element = getInput(scanner);
            set.add(element);
        }

        // Print the set
        System.out.println("Set elements: " + set);

        System.out.print("Enter the element to remove: ");
        Object elementToRemove = getInput(scanner);

        // Remove the specified element from the set
        if (set.remove(elementToRemove)) {
            System.out.println("Element removed successfully.");
        } else {
            System.out.println("Element not found in the set.");
        }

        // Print the updated set
        System.out.println("Updated set elements: " + set);

        // Close the scanner
        scanner.close();
    }

    private static Object getInput(Scanner scanner) {
        if (scanner.hasNextInt()) {
            return scanner.nextInt();
        } else if (scanner.hasNextDouble()) {
            return scanner.nextDouble();
        } else {
            return scanner.next();
        }
    }
}
```

OUTPUT:



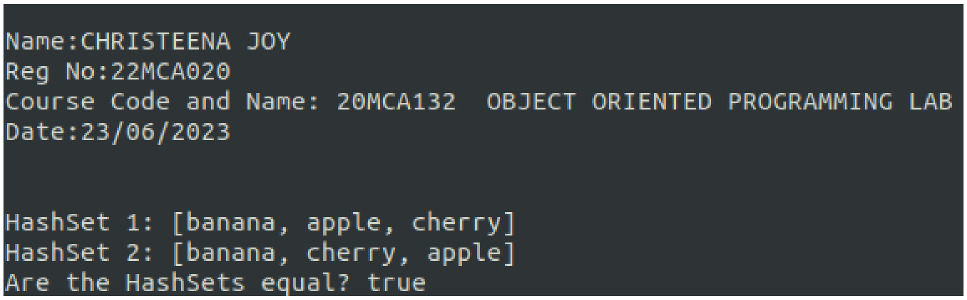
```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:23/06/2023


Enter the number of elements to add: 3
Enter the elements:
apple
orange
banana
Set elements: [apple, orange, banana]
Enter the element to remove: orange
Element removed successfully.
Updated set elements: [apple, banana]
```

16. Write a Java program to compare two hash set

CODE:

```
import java.util.HashSet;

public class HashSetComparison {
    public static void main(String[] args) {
    System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:23/06/2023\n\n");
        // Create the first HashSet
        HashSet<String> set1 = new HashSet<>();
        set1.add("apple");
        set1.add("banana");
        set1.add("cherry");

        // Create the second HashSet
        HashSet<String> set2 = new HashSet<>();
        set2.add("banana");
        set2.add("cherry");
        set2.add("apple");

        // Compare the HashSets
        boolean isEqual = set1.equals(set2);

        // Display the result
        System.out.println("HashSet 1: " + set1);
        System.out.println("HashSet 2: " + set2);
        System.out.println("Are the HashSets equal? " + isEqual);
    }
}
```

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:23/06/2023


HashSet 1: [banana, apple, cherry]
HashSet 2: [banana, cherry, apple]
Are the HashSets equal? true
```

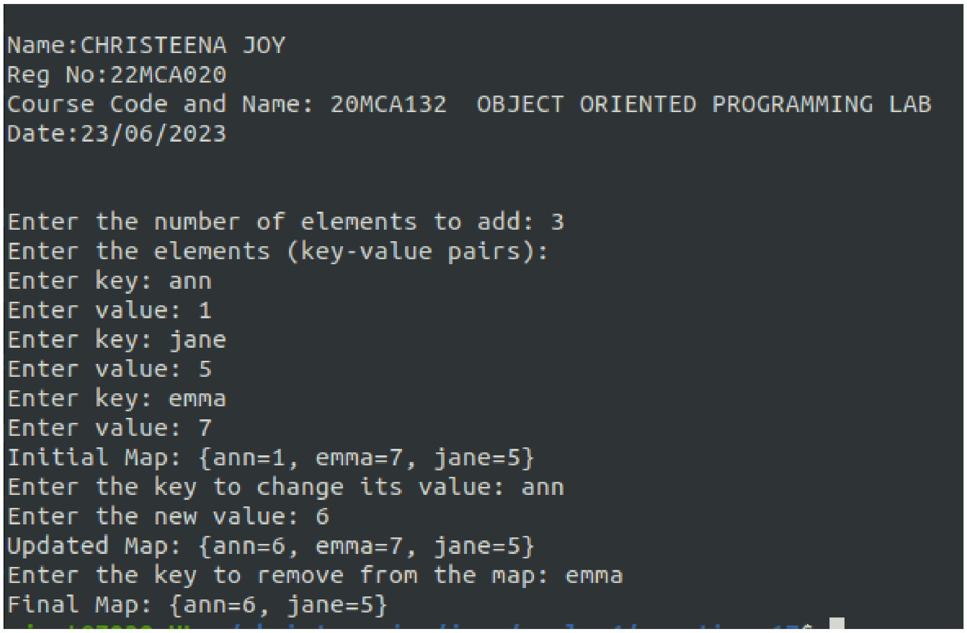
17. Program to demonstrate the working of Map interface by adding, changing and removing Elements.

CODE:

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class MapDemo {
    public static void main(String[] args) {
        System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and
Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:23/06/2023\n\n");
        // Create a Map
        Map<String, Integer> map = new HashMap<>();

        // Create a Scanner for user input
        Scanner scanner = new Scanner(System.in);

        // Get the number of elements to add
        System.out.print("Enter the number of elements to add: ");
        int numElements = scanner.nextInt();

        // Get the elements from the user
        System.out.println("Enter the elements (key-value pairs):");
        for (int i = 0; i < numElements; i++) {
            System.out.print("Enter key: ");
            String key = scanner.next();

            System.out.print("Enter value: ");
            int value = scanner.nextInt();

            // Add the element to the map
            map.put(key, value);
        }

        // Display the initial map
        System.out.println("Initial Map: " + map);

        // Get the key to change its value
        System.out.print("Enter the key to change its value: ");
        String keyToChange = scanner.next();

        // Check if the key exists in the map
        if (map.containsKey(keyToChange)) {
            System.out.print("Enter the new value: ");
            int newValue = scanner.nextInt();

            // Change the value of the key
            map.put(keyToChange, newValue);
        } else {
            System.out.println("Key does not exist in the map.");
        }

        // Display the updated map
        System.out.println("Updated Map: " + map);

        // Get the key to remove from the map
        System.out.print("Enter the key to remove from the map: ");
        String keyToRemove = scanner.next();

        // Remove the key if it exists in the map
        if (map.containsKey(keyToRemove)) {
            map.remove(keyToRemove);
        } else {
            System.out.println("Key does not exist in the map.");
        }
```

```
        // Display the final map
        System.out.println("Final Map: " + map);

        // Close the scanner
        scanner.close();
    }
}
```

OUTPUT:



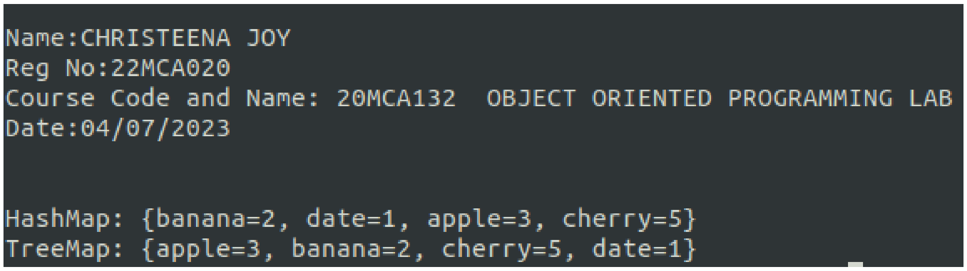
```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:23/06/2023


Enter the number of elements to add: 3
Enter the elements (key-value pairs):
Enter key: ann
Enter value: 1
Enter key: jane
Enter value: 5
Enter key: emma
Enter value: 7
Initial Map: {ann=1, emma=7, jane=5}
Enter the key to change its value: ann
Enter the new value: 6
Updated Map: {ann=6, emma=7, jane=5}
Enter the key to remove from the map: emma
Final Map: {ann=6, jane=5}
```

18. Program to Convert HashMap to TreeMap

CODE:

```
import java.util.HashMap;
import java.util.Map;
import java.util.TreeMap;

public class HashMapToTreeMapDemo {
   public static void main(String[] args) {
   System.out.println("\nName:CHRISTEENA JOY\nReg No:22MCA020\nCourse Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB\nDate:04/07/2023\n\n");
      // Create a HashMap
      HashMap<String, Integer> hashMap = new HashMap<>();
      hashMap.put("apple", 3);
      hashMap.put("banana", 2);
      hashMap.put("cherry", 5);
      hashMap.put("date", 1);

      // Convert HashMap to TreeMap
      TreeMap<String, Integer> treeMap = new TreeMap<>(hashMap);

      // Display the HashMap
      System.out.println("HashMap: " + hashMap);

      // Display the TreeMap
      System.out.println("TreeMap: " + treeMap);
   }
}
```

OUTPUT:



```
Name:CHRISTEENA JOY
Reg No:22MCA020
Course Code and Name: 20MCA132  OBJECT ORIENTED PROGRAMMING LAB
Date:04/07/2023


HashMap: {banana=2, date=1, apple=3, cherry=5}
TreeMap: {apple=3, banana=2, cherry=5, date=1}
```