IRC_JAVA_OOP_CC_2_Slot_2

Test Summary

- No. of Sections: 1
- No. of Questions: 5
- Total Duration: 60 min

Section 1 - Coding

Section Summary

- No. of Questions: 5
- Duration: 60 min

Additional Instructions:

None

Q1. Write a program by creating a class Bicycle as a base class with a number of gears and speed of bicycle as integer attributes and create a class called MountainBike, a derived class that extends Bicycle class with an attribute seat height as an integer. Create a Test class to run the program and obtain the output in the console.
Note: Override toString() method to display the details of the bicycle.

Input Format

To get 3 integers from the user (Number of gears, Speed of bicycle, and Seat height).

Output Format

To display the desired output from the test class.

Constraints

integers only.

Sample Input

```
2 90 40
```

Sample Output

```
No of gears are 2
speed of bicycle is 90
seat height is 40
```

Sample Input

```
3 60 20
```

Sample Output

```
No of gears are 3
speed of bicycle is 60
seat height is 20
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. Given a Book class and the Main class, write a MyBook class that does the following:
Inherits from Book
Has a parameterized constructor taking these 3 parameters:

- string title
- string author
- int price

Implements the Book class' abstract display() method so it prints the title, author, and price.

Input Format

The Main class creates a Book object and calls the MyBook class constructor (passing it the necessary arguments). It then calls the display method on the Book object.

Output Format

The void display() method should print and label the respective title, author, and price of the MyBook object's instance (with each value on its own line) .

Constraints

Strings and integers only.

Sample Input

Sample Output

```
love
jack
300
```

```
Title: love
Author: jack
Price: 300
```

Sample Input **Sample Output**

```
stars
juno
150
```

```
Title: stars
Author: juno
Price: 150
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. A company maintains a database that has the details of all the employees. There are two levels of employees where level 1 is the top management having salary more than 100 dollars and level 2 is the staffs who are getting a salary less than 100 dollars. Create a class named Employee with empId and salary as attributes. Create another class empLevel that extends employee and categorizes the employee into various levels.

Input Format

The input should contain only the employee id and salary of the employee separated by space.
Employee id should be of integer type and salary float type.

Output Format

The output of the program must display the employee id, salary, and level of the employee one below the other in the same order.

Sample Input **Sample Output**

```
253 5.6
```

```
253
5.6
2
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. Create an abstract class **Shape** with length, width, radius, 3 sides as data members and two abstract methods to calculate area and perimeter. Create constructors and getter setters.

Create four classes **Square**, **Rectangle**, **Circle** and **Triangle.** Extend all the classes from **Shape** directly**.** Complete the abstract method to calculate area and perimeter in the derived classes.

Get a single character and suitable values from user to calculate area and perimeter.

Input Format

S or R or C or T in first line (S represents Square, R represents Rectangle, C represents Circle and T represents Triangle)
Enter one or two input based on Shape (1 input for Square and Circle, 2 inputs for Rectangle and 3 inputs for Triangle)

Output Format

Perimeter or Circumference
Area

Sample Input **Sample Output**

```
S
5
```

```
Perimeter : 20.00
Area : 25.00
```

Sample Input **Sample Output**

```
R
3
4
```

```
Perimeter : 14.00
Area : 12.00
```

Sample Input **Sample Output**

```
C
7
```

```
Circumference : 43.98
Area : 153.94
```

Sample Input **Sample Output**

```
T
3
```

```
Perimeter : 12.00
Area : 6.00
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q5. Write a program such that it should have three classes, Person and Student and the Main class where Person is the base class and Student is the derived class.
Student class should have the following:
A Student class constructor, which has parameters:
A string, firstName.
A string, lastName.
An integer, id.
An integer array of test scores.
A char calculate() method that calculates a Student object's average and returns the grade character representative of their calculated average:

Grading Scale

Letter	Average (a)
O	$90 \le a \le 100$
E	$80 \le a < 90$
A	$70 \le a < 80$
P	$55 \le a < 70$
D	$40 \le a < 55$
T	$a < 40$

Input Format

Input to get :
A string, firstName in the first line.
A string, lastName in the second line.
An integer, id in the third line.
A number of test scores in the fourth line.
An integer array of test scores is separated by a single space in the last line.

Output Format

Display the output as shown in the sample output.

Sample Input

```
Shruthi
Ramesh
50
5
```

Sample Output

```
Name: Ramesh Shruthi
ID: 50
Grade: E
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - Coding

Q1

Test Case

Input

```
3 50 12
```

Output

```
No of gears are 3
speed of bicycle is 50
seat height is 12
```

Weightage - 10

Input

```
1 40 10
```

Output

```
No of gears are 1
speed of bicycle is 40
seat height is 10
```

Weightage - 10

Input

```
3 50 28
```

Output

```
No of gears are 3
speed of bicycle is 50
seat height is 28
```

Weightage - 10

Input

```
1 44 22
```

Output

```
No of gears are 1
speed of bicycle is 44
seat height is 22
```

Weightage - 10

Input

```
4 60 30
```

Output

```
No of gears are 4
speed of bicycle is 60
seat height is 30
```

Weightage - 10

Input

```
2 69 30
```

Output

```
No of gears are 2
speed of bicycle is 69
seat height is 30
```

Weightage - 10

Input

```
4 80 22
```

Output

```
No of gears are 4
speed of bicycle is 80
seat height is 22
```

Weightage - 20

Input

```
1 79 43
```

Output

```
No of gears are 1
speed of bicycle is 79
seat height is 43
```

Weightage - 20

Sample Input

```
2 90 40
```

Sample Output

```
No of gears are 2
speed of bicycle is 90
seat height is 40
```

Sample Input

```
3 60 20
```

Sample Output

```
No of gears are 3
speed of bicycle is 60
seat height is 20
```

Solution

```
import java.util.Scanner;
class Bicycle
{
    public int gear;
    public int speed;

    public Bicycle(int gear, int speed)
    {
        this.gear = gear;
        this.speed = speed;
    }

    public void applyBrake(int decrement)
    {
        speed -= decrement;
    }

    public void speedUp(int increment)
    {
        speed += increment;
    }

    // toString() method to print info of Bicycle
    public String toString()
    {
        return("No of gears are "+gear
                +"\n"
                + "speed of bicycle is "+speed);
    }
}

class MountainBike extends Bicycle
{

    public int seatHeight;

    public MountainBike(int gear,int speed,
```

```
                        int startHeight)
    {
        super(gear, speed);
        seatHeight = startHeight;
    }

    public void setHeight(int newValue)
    {
        seatHeight = newValue;
    }

    @Override
    public String toString()
    {
        return (super.toString()+
                "\nseat height is "+seatHeight);
    }

}

class Test
{
    public static void main(String args[])
    {
         int gear,speed,startHeight;
         Scanner sc=new Scanner(System.in);
         gear=sc.nextInt();
         speed=sc.nextInt();
         startHeight=sc.nextInt();
        MountainBike mb = new MountainBike(gear,speed,startHeight);
        System.out.println(mb.toString());

    }
}
```

Q2

Test Case

Input

```
threestates
chetan
200
```

Output

```
Title: threestates
Author: chetan
Price: 200
```

Weightage - 10

Input

```
blowmind
jenny
400
```

Output

```
Title: blowmind
Author: jenny
Price: 400
```

Weightage - 10

Input

```
happy
arora
300
```

Output

```
Title: happy
Author: arora
Price: 300
```

Weightage - 10

Input

```
pathetic
iqbal
600
```

Output

```
Title: pathetic
Author: iqbal
Price: 600
```

Weightage - 10

Input

```
moms
prema
400
```

Output

```
Title: moms
Author: prema
Price: 400
```

Weightage - 10

Input

```
lightning
nerosa
700
```

Output

```
Title: lightning
Author: nerosa
Price: 700
```

Weightage - 10

Input

```
today
sam
500
```

Output

```
Title: today
Author: sam
Price: 500
```

Weightage - 20

Input

```
global
kalam
250
```

Output

```
Title: global
Author: kalam
Price: 250
```

Weightage - 20

Sample Input

```
love
jack
300
```

Sample Output

```
Title: love
Author: jack
Price: 300
```

Sample Input

```
stars
juno
150
```

Sample Output

```
Title: stars
Author: juno
Price: 150
```

Solution

```
import java.util.Scanner;
class Main {

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

```
        Scanner sc=new Scanner(System.in);
        String title =sc.nextLine();
        String author =sc.nextLine();
        int price = sc.nextInt();

        Book book = new MyBook(title, author, price);
        book.display();
    }
}
abstract class Book {
    String title;
    String author;

    Book(String title, String author) {
        this.title = title;
        this.author = author;
    }

    abstract void display();
}
class MyBook extends Book
{
    int price;
    MyBook(String title,String author,int price)
    {
        super(title,author);
        this.price=price;
    }

    void display()
    {
        System.out.println("Title: "+title);
        System.out.println("Author: "+author);
        System.out.println("Price: "+price);

    }
}
```

Q3

Test Case

Input	Output
156 888	156 888.0 1

Weightage - 20

Input	Output
454 100	454 100.0 2

Weightage - 20

Input	Output
485 46	485 46.0 2

Weightage - 10

Input

```
45  100.2
```

Output

```
45
100.2
1
```

Weightage - 20

Input

```
9658  88.000
```

Output

```
9658
88.0
2
```

Weightage - 20

Input

```
54  96
```

Output

```
54
96.0
2
```

Weightage - 10

Sample Input

```
253  5.6
```

Sample Output

```
253
5.6
2
```

Solution

```
import java.util.Scanner;
class Employee
{
    public int empId;
    public float slry;

    public Employee(int empId, float slry)
    {
        this.empId = empId;
        this.slry = slry;
    }
    public String toString()
    {
        return (empId + "\n" + slry);
    }
}
class empLevel extends Employee
{
    public int level;
    public empLevel(int empId, float slry)
    {
        super(empId, slry);
        if (slry>100)
            level = 1;
        else
            level = 2;
```

```
    }
    public String toString()
    {
        return (super.toString() + "\n" + level);
    }
}
class empMain
{
    public static void main(String args[])
    {
        int emplId;
        float slry;
        Scanner in = new Scanner(System.in);
        emplId = in.nextInt();
        slry = in.nextFloat();
        empLevel el = new empLevel(emplId,slry);
        System.out.println(el.toString());
    }
}
```

```
using System;

namespace ConsoleApp2
{
    class Employee
    {
        public int empId;
        public float slry;

        public Employee(int empId, float slry)
        {
            this.empId = empId;
            this.slry = slry;
        }

        public override string ToString()
        {
            return (empId + "\n" + slry.ToString("F1"));
        }

    }
    class empLevel : Employee
    {
        public int level;
        public empLevel(int empId, float slry) : base(empId, slry)
        {

            if (slry > 100)
                level = 1;
            else
                level = 2;
        }
        public override string ToString()
        {
            return (base.ToString() + "\n" + level);
        }

}


class Program
{

    static void Main(string[] args)
```

```
        {
            int emplId;
            float slry;
            string[] tokens = Console.ReadLine().Split();


            emplId = int.Parse(tokens[0]);
            slry = float.Parse(tokens[1]);
            empLevel el = new empLevel(emplId, slry);
            Console.WriteLine(el.ToString());
        }
    }
}
```

Q4

Test Case

Input

```
S
8
```

Output

```
Perimeter : 32.00
Area : 64.00
```

Weightage - 25

Input

```
R
10
15
```

Output

```
Perimeter : 50.00
Area : 150.00
```

Weightage - 25

Input

```
C
10
```

Output

```
Circumference : 62.83
Area : 314.16
```

Weightage - 25

Input

```
T
9
10
11
```

Output

```
Perimeter : 30.00
Area : 42.43
```

Weightage - 25

Sample Input

```
S
5
```

Sample Output

```
Perimeter : 20.00
Area : 25.00
```

Sample Input

```
R
3
4
```

Sample Output

```
Perimeter : 14.00
Area : 12.00
```

Sample Input

```
C
7
```

Sample Output

```
Circumference : 43.98
Area : 153.94
```

Sample Input

```
T
3
4
5
```

Sample Output

```
Perimeter : 12.00
Area : 6.00
```

Solution

```
import java.util.Scanner;
abstract class Shape {
    private double length, width; //square and rectangle
    private double radius; //circle
    private double side1, side2, side3; //triangle

//set length and width
public void setLengthAndWidth(double l, double w){
    length = l;
    width = w;
}

//return length
public double getLength(){
    return length;
}

//return width
public double getWidth(){
    return width;
}

//set radius
public void setRadius(double r){
    radius = r;
}

//return radius
public double getRadius(){
    return radius;
}


//set sides of triangle
public void setSides(double s1, double s2, double s3){
    side1 = s1;
    side2 = s2;
    side3 = s3;
}

//return side1
public double getSide1(){
    return side1;
}

//return side2
public double getSide2(){
    return side2;
}
```

```
    //return side3
    public double getSide3(){
        return side3;
    }


    public abstract String toString();  //return String representation of the class
    public abstract double area();  //calculate area
    public abstract double perimeter(); //calculate perimeter
}

//class Circle
 class Circle extends Shape{
    public Circle(double radius){
        setRadius(radius);
    }

    public Circle(){
        //do nothing
    }

    //calculate area of circle
    @Override
    public double area(){
        return Math.PI * Math.pow(getRadius(), 2);
    }

    //calculate circumference of circle
    @Override
    public double perimeter(){
        return 2 * Math.PI * getRadius();
    }


    //return String representation of area
    @Override
    public String toString() {
        return String.format("Circumference : %.2f\nArea : %.2f\n", perimeter(), area());
    }
}


 class Rectangle extends Shape{

public Rectangle(double length, double width){
    setLengthAndWidth(length, width);
}

public Rectangle(){
    //do nothing
}

//calculate area
@Override
public double area(){
    return getLength() * getWidth();
}

//calculate perimeter
@Override
public double perimeter(){
    return 2 * (getLength() + getWidth());
}
```

```
    //return String representation of area and perimeter
    @Override
    public String toString(){
        return String.format("Perimeter : %.2f\nArea : %.2f\n", perimeter(), area());
    }
}

 class Square extends Shape{

    public Square(double l1, double l2){
        setLengthAndWidth(l1, l2);
    }

    public Square(){
        //do nothing
    }

    //calculate area
    @Override
    public double area(){
        return getLength() * getWidth();
    }

    //calculate perimeter
    @Override
    public double perimeter(){
        return 2 * (getLength() + getWidth());
    }

    //return string representation of perimeter and area
    @Override
    public String toString(){
        return String.format("Perimeter : %.2f\nArea : %.2f\n",perimeter(),area());
    }
}

 class Triangle extends Shape{
    public Triangle(double side1, double side2, double side3){
        setSides(side1, side2, side3);
    }

    public Triangle(){
        //do nothing
    }

    @Override
    public double area(){
        double p = (1/2.0) * (getSide1() + getSide2() + getSide3());
        return Math.sqrt(p * (p-getSide1()) * (p-getSide2()) * (p-getSide3()));
    }

    //calculate perimeter
    @Override
    public double perimeter(){
        return getSide1() + getSide2() + getSide3();
    }

    //return String representation of area and perimeter
    @Override
    public String toString(){
        return String.format("Perimeter : %.2f\nArea : %.2f", perimeter(), area());
    }
}

 class Main {
```

```
public static void main(String[] args){
    Scanner input = new Scanner(System.in);
    double radius, length, width;

    //System.out.println("Select a shape");
    //System.out.print("1.Square\n2.Rectangle\n3.Circle\n4.Triangle\n? ");
    char choice = input.nextLine().charAt(0);
    //System.out.println(choice);
    switch(choice){
        case 'S':case 's':
            //System.out.println("Enter the length of one side of the square");
            length = Double.parseDouble(input.nextLine());
            Square square = new Square(length, length);
            System.out.println(square.toString());
            break;

        case 'R': case 'r':
        //  System.out.println("Enter the length and width of the rectangle");
        //  System.out.print("Length : ");
            length = Double.parseDouble(input.nextLine());
        //  System.out.print("Width : ");
            width = Double.parseDouble(input.nextLine());
            Rectangle rectangle = new Rectangle(length, width);
            System.out.println(rectangle.toString());
            break;

    case 'C': case 'c':
    //  System.out.println("Enter radius of the circle");
        radius = Double.parseDouble(input.nextLine());
        Circle circle = new Circle(radius);
        System.out.println(circle.toString());
        break;

    case 'T': case't':
    //  System.out.println("Enter the length of the sides of the triangle");
    //  System.out.print("side1: ");
        double s1 = Double.parseDouble(input.nextLine());
    //  System.out.print("side2: ");
        double s2 = Double.parseDouble(input.nextLine());
    //  System.out.print("side3: ");
        double s3 = Double.parseDouble(input.nextLine());
        Triangle triangle = new Triangle(s1, s2, s3);
        System.out.println(triangle.toString());
        break;
            default:
            //  System.out.println("Invalid choice");
        }
        input.close();

    }
}
```

Q5

Test Case

Input	Output
Raj Mohan 45 6	Name: Mohan Raj ID: 45 Grade: E

Weightage - 20

Input

```
Rinny
Jo
40
4
```

Output

```
Name: Jo Rinny
ID: 40
Grade: D
```

Weightage - 20

Input

```
Prem
Kumar
32
5
```

Output

```
Name: Kumar Prem
ID: 32
Grade: E
```

Weightage - 20

Input

```
Billa
Babu
6
5
```

Output

```
Name: Babu Billa
ID: 6
Grade: P
```

Weightage - 20

Input

```
Nethra
Sampath
40
6
```

Output

```
Name: Sampath Nethra
ID: 40
Grade: E
```

Weightage - 20

Sample Input

```
Shruthi
Ramesh
50
5
```

Sample Output

```
Name: Ramesh Shruthi
ID: 50
Grade: E
```

Solution

```
import java.io.*;
import java.util.*;
import java.util.Scanner;
class Person {
    protected String firstName;
    protected String lastName;
    protected int idNumber;
    Person(String firstName, String lastName, int identification){
        this.firstName = firstName;
        this.lastName = lastName;
        this.idNumber = identification;
    }

    public void printPerson(){
         System.out.println(
                "Name: " + lastName + " " + firstName
            +   "\nID: " + idNumber);
    }

}
```

```
class Student extends Person{
    private int[] testScores;

      public Student(String firstName, String lastName, int id, int[] testScores) {
         super(firstName, lastName, id);
              this.testScores=testScores;

        }
         char calculate()
         {
            int i,sum=0,avg;
            char grade;
         for(i=0; i<testScores.length; i++)
             sum+=testScores[i];
         avg=sum/testScores.length;

         if(avg<=100 && avg>=90)
            grade='O';
         else if(avg>=80&&avg<90)
            grade='E';
         else if(avg>=70&&avg<80)
            grade='A';
          else if(avg>=55&&avg<70)
             grade='P';
          else if(avg>=40&&avg<55)
             grade='D';
          else
             grade='T';

          return grade;
     }}
class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String firstName = scan.next();
        String lastName = scan.next();
        int id = scan.nextInt();
        int numScores = scan.nextInt();
        int[] testScores = new int[numScores];
        for(int i = 0; i < numScores; i++){
            testScores[i] = scan.nextInt();
        }
        scan.close();
        Student s = new Student(firstName, lastName, id, testScores);
        s.printPerson();
        System.out.println("Grade: " + s.calculate());
    }}
```