DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

Worksheet 2

Student Name : Anand Kashyap **UID:** 24BCS11720
Branch: BE CSE **Section/Group:** 707 (B)
Semester: 4 **Date of Performance:** 18/2/26
Subject Name: OBJECT ORIENTED PROGRAMMING USING JAVA **Subject Code:** 24CSH-207

Aim:

1. To design and implement a system for managing and calculating employee salary details.
2. To design and implement a program to calculate the area of different geometric shapes.

Requirements (Software): Web Browser (for accessing HackerRank), Operating System

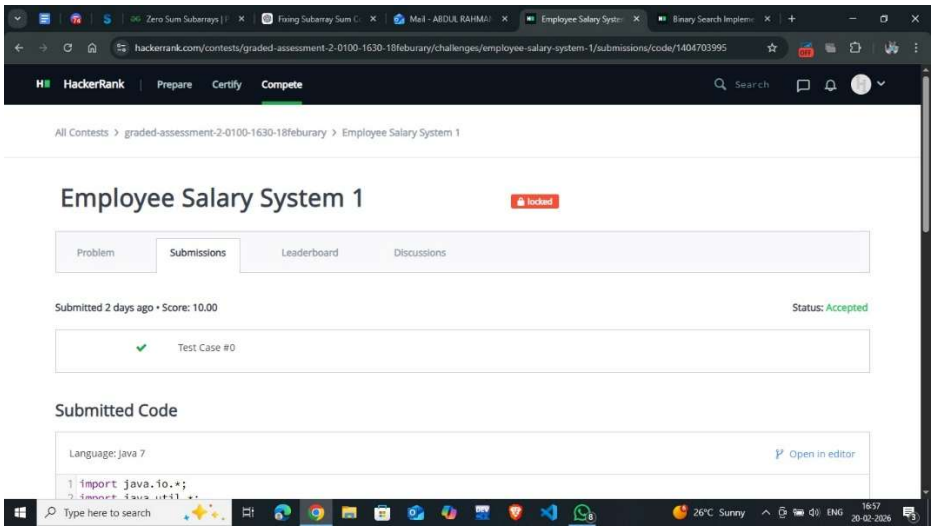
Code:

```
// Name : Anand Kashyap
// UID  : 24BCS11720

} import java.util.*;
import java.io.*;

class Employee {
    protected String name;
    protected int id;
    protected double basicSalary;

    public Employee(String name, int id, double basicSalary) {
        this.name = name;
        this.id = id;
        this.basicSalary = basicSalary;
    }

    public double calculateSalary() {
        return basicSalary;
    }
```

```
class FullTimeEmployee extends Employee {
    private double hra;
    private double da;

    public FullTimeEmployee(String name, int id, double basicSalary, double hra, double da) {
        super(name, id, basicSalary);
        this.hra = hra;
        this.da = da;
    }

    @Override
    public double calculateSalary() {
        return basicSalary + hra + da;
    }
}

class PartTimeEmployee extends Employee {
    private int hoursWorked;
    private double ratePerHour;

    public PartTimeEmployee(String name, int id, int hoursWorked, double ratePerHour) {
        super(name, id, 0);
        this.hoursWorked = hoursWorked;
        this.ratePerHour = ratePerHour;
    }

    @Override
    public double calculateSalary() {
        return hoursWorked * ratePerHour;
    }
}

public class Solution {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Input example (as per typical HR problems)
        String name = sc.nextLine();
        int id = sc.nextInt();
        int type = sc.nextInt(); // 1 = FullTime, 2 = PartTime

        if (type == 1) {
```

```
            double basic = sc.nextDouble();
            double hra = sc.nextDouble();
            double da = sc.nextDouble();

            FullTimeEmployee emp = new FullTimeEmployee(name, id, basic, hra, da);
            System.out.printf("%.2f", emp.calculateSalary());

        } else {
            int hours = sc.nextInt();
            double rate = sc.nextDouble();

            PartTimeEmployee emp = new PartTimeEmployee(name, id, hours, rate);
            System.out.printf("%.2f", emp.calculateSalary());
        }

        sc.close();
    }
}
```

OUTPUT:



Q- 2 To design and implement a program to calculate the area of different geometric shapes.

Code:

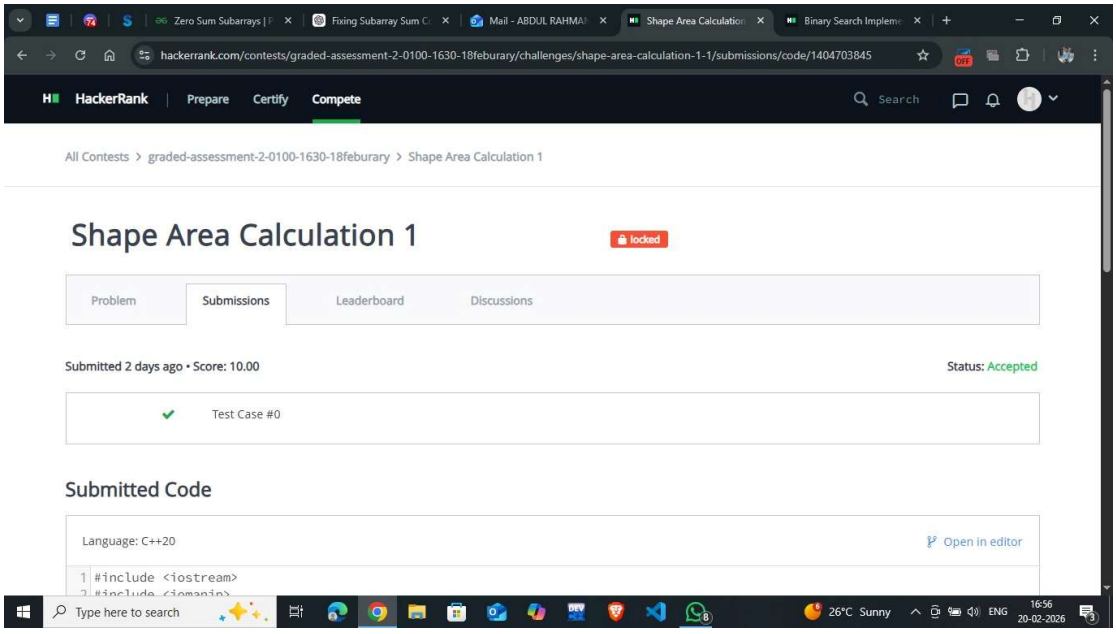
```
// Name : Anand Kashyap
// UID  : 24BCS11720

#include <bits/stdc++.h>

using namespace std;
```

```
class Shape {
public:
    virtual double area() = 0;
};

class Rectangle : public Shape {
    double length, breadth;
public:
    Rectangle(double l, double b) {
        length = l;
        breadth = b;
    }
    double area() {
        return length * breadth;
    }
};

class Circle : public Shape {
    double radius;
public:
    Circle(double r) {
        radius = r;
    }
    double area() {
        return 3.14159 * radius * radius;
    }
};

class Triangle : public Shape {
    double base, height;
public:
```

```
Triangle(double b, double h) {
        base = b;
        height = h;
    }
    double area() {
        return 0.5 * base * height;
    }
};

int main() {
    int choice;
    cin >> choice;
    // 1 = Rectangle, 2 = Circle, 3 = Triangle

    Shape* s;

    if (choice == 1) {
        double l, b;
        cin >> l >> b;
        s = new Rectangle(l, b);
    }
    else if (choice == 2) {
        double r;
        cin >> r;
        s = new Circle(r);
    }
    else if (choice == 3) {
        double base, height;
        cin >> base >> height;
        s = new Triangle(base, height);
```

```
cout << fixed << setprecision(2) << s->area();
    return 0;
}
```

OUTPUT:



Learning Outcomes:

After completing this experiment, the student will be able to:

1. Understand how to structure a program using functions/classes to solve real-world problems.
2. Learn to handle user input and perform mathematical calculations programmatically.
3. Improve logical thinking by applying conditions and formulas correctly.
4. Gain confidence in implementing problem statements in any programming language.
5. Develop the ability to write modular and reusable code for different problem domains.