Programming Test Results (With Test Cases)

Result Summary

Field	Value
Test ID	39182
Student ID	29195
Programs (with test cases)	3
Total Test Cases	9
Test Cases Passed	9
Fully Passed Programs	3
Partially Passed Programs	0
Failed Programs	0
Overall % (with test cases)	100.00%
Grade	Outstanding

Programs With Test Cases

#	Program Name	Total TC	Passed	Success Rate	Score	Submitted At	Attempts
1	StudentTracker	4	4	100.0%	10	15/11/2025, 13:14:21	0
2	EmployeeSalaryDemo	2	2	100.0%	10	15/11/2025, 12:26:34	0
3	RectangleELC	3	3	100.0%	10	15/11/2025, 11:51:46	0

Program Details (With Test Cases)

Program 1: StudentTracker

Field	Value
Program No.	1
Program Name	StudentTracker
Description	Person (Base Class) Represents a generic person with basic information. Instance Variables:

Field	Value
	name (String) id (int) Constructor: Initializes name and id. Ø=Ý9 Student (Inherits from Person) Represents a student with academic effort tracking. Instance Variables: hardWorkScore (int) !’ (0–100) taskCompleted (int) !’ (number of assignments out of 10) Constructor: Calls super() and initializes additional fields. Ø=Ý9 PlacementCandidate (Inherits from Student) Represents a final-year student evaluated for placement. Instance Variables: classAttendance (double) !’ percentage (0–100) labAttendance (double) !’ percentage (0–100) Constructor: Uses constructor chaining to initialize all fields. Method: void evaluateCandidate(int choice): menu using switch-case or if else : 1.Show all scores 2.Calculate placement probability 3.Show eligibility status (based on thresholds) 4.Exit Placement Probability Logic: probability = (hardWorkScore * 0.3 + taskCompleted * 5 + classAttendance * 0.2 + labAttendance * 0.2) [Note :---- Eligibility Rules: Hard work score "e 70 Task completed "e 7 Class & Lab attendance "e 75%] Take an ELC class StudentTracker and instantiate the PlacementCandidate and invoke the evaluateCandidate method .

Field	Value
	Sample Input : ------------------- Name: virat ID: 101 HardWorkScore: 85 TaskCompleted: 9 ClassAttendance: 80.0 LabAttendance: 85.0 Sample Output : --------------------- Candidate: VIRAT (ID: 101) === Placement Evaluation Menu === 1. View All Scores 2. Check Placement Probability 3. Check Placement Eligibility 4. Exit ------------------------ Enter your choice: 1 Hard Work Score: 85 Tasks Completed: 9/10 Class Attendance: 80.0% Lab Attendance: 85.0% ------------------------ Enter your choice: 2 Estimated Placement Probability: 92.5% ------------------------ Enter your choice: 3 You are ELIGIBLE for placement! ------------------------ Enter your choice: 4 Exiting...
Constraints	
Sample Input	virat 101 85 9 80.0 85.0 1
Sample Output	Hard Work Score: 85 Tasks Completed: 9/10 Class Attendance: 80.0% Lab Attendance: 85.0%
Explanation	-
Language(s)	java
Total Test Cases	4

Field	Value
Test Cases Passed	4
Test Cases Failed	0
Success Rate	100.0%
Score (0–10)	10
Attempts	0
Submitted At	15/11/2025, 13:14:21

Code

```
import java.util.*;
public class  StudentTracker{
    public static void main(String [] args){
        Scanner sc = new Scanner(System.in);
        String name = sc.nextLine();
        int id =Integer.parseInt(sc.nextLine());
        int hardWorkScore = Integer.parseInt(sc.nextLine());
        int taskCompleted = Integer.parseInt(sc.nextLine());
        double clas = Double.parseDouble(sc.nextLine());
        double lab = Double.parseDouble(sc.nextLine());
        PlacementCandidate p = new
PlacementCandidate(name,id,hardWorkScore,taskCompleted,clas,lab);
        int c = Integer.parseInt(sc.nextLine());
        p.evalutate(c);
    }
}
class Person{
    String name;
    int id;
    Person(String name, int id){
        this.name= name;
        this.id= id;
    }
}
class Student extends Person{
  int hardWorkScore;
  int taskCompleted;
  Student(String name,int id,int hardWorkScore,int taskCompleted){
    super(name,id);
    this.hardWorkScore = hardWorkScore;
    this.taskCompleted=taskCompleted;
  }
}
class PlacementCandidate extends Student{
 double Percentage;
 double labAttendance;
 PlacementCandidate(String name,int id,int hardWorkScore,int taskCompleted,double
Percentage,double labAttendance){
    super(name, id,hardWorkScore,taskCompleted);
    this.Percentage=Percentage;
    this.labAttendance = labAttendance;
 }
 public void evalutate(int choice){
    if(choice ==1){
System.out.println("Hard Work Score: "+hardWorkScore);
System.out.println("Tasks Completed: "+taskCompleted+"/10");
System.out.println("Class Attendance: "+Percentage+"%");
System.out.println("Lab Attendance: "+labAttendance+"%");
    }
    else if(choice==2){
        double per=(hardWorkScore*0.3+taskCompleted*5+Percentage*0.2+labAttendance*0.2);
        System.out.printf("Estimated Placement Probability: %.2f%%",per);

    }
    else{
        System.out.println("You are NOT eligible for placement.");
    }
 }
}
```

Program 2: EmployeeSalaryDemo

Field	Value
Program No.	2
Program Name	EmployeeSalaryDemo
Description	Create a class Employee (Business Logic Class) Non static Fields : employeeNumber : private-int employeeName : private-String employeeSalary:private -double Create a parameterized constructor to initialilize all the fields. Create a pair of setter and getter methods for all the Non static fields. Methods : --------- 1) Method Name : getEmployeeDesignation() Parameter : double salary [Here method should receive updated salary] Return Type : String Access modifier : public In this method, Take the updated salary in the parameter variable and return the employee designation based on the following criteria : a) If updatedSalary is greater than 120000, return Employee is a HR Manager. b) If updatedSalary is greater than 90000, return Employee is a Developer. c) If updatedSalary is greater than 60000, return Employee is a Designer. d) In the else part, return Employee is a Tester. Take toString() method to print Employee Object properties. Create an ELC class EmployeeDemo which contains main method, with the help of Scanner class take the input from the user to initialize the non static field through parameterized constructor. Print the employee details using toString() method. Take the salary increment amount from the user using Scanner class, update the employee salary using setter and getter. Print the employee data with updated salary. Pass this updated salary to getEmployeeDesignation() method to get and print the Employee Designation as per below output.
Constraints	
Sample Input	101 Ravi 80000 10000
Sample Output	Updated Salary = 90000.0 !' Employee is a Developer
Explanation	-
Language(s)	java

Field	Value
Total Test Cases	2
Test Cases Passed	2
Test Cases Failed	0
Success Rate	100.0%
Score (0–10)	10
Attempts	0
Submitted At	15/11/2025, 12:26:34

Code

```
import java.util.*;
public class EmployeeDemo{
    public static void main(String [] args){
        Scanner sc = new Scanner(System.in);
        int eNO =Integer.parseInt(sc.nextLine());
        String name = sc.nextLine();
        double Salary = Double.parseDouble(sc.nextLine());
        double update = Double.parseDouble(sc.nextLine());
        Employee e = new Employee(eNO,name,Salary);
        System.out.println(e.getEmployeeDesigntion());
        e.setSalary(update);
      //  System.out.println();
        System.out.println(e.toString());
    }
}
class Employee{
    private int emNo;
    private String emName;
    private double emSalary;

    Employee(int emNo,String emName,double emSalary){
        this.emNo=emNo;
        this.emName=emName;
        this.emSalary=emSalary;
    }
    public String getEmployeeDesigntion(){
       return "Employee Details:"+"\n"+"Employee Number: "+this.emNo+"\n"+"Employee Name:
"+this.emName+"\n"+"Employee Salary: "+this.emSalary;
    }
    public void setSalary(double Salary){
        this.emSalary+=Salary;
    }
    public String toString(){
String Posi="0";
        if(this.emSalary>=120000){
         Posi="Employee is a HR Manager.";
        }
        else if(this.emSalary>=90000){
            Posi = "Employee is a Developer.";
        }
        else if(this.emSalary>=60000){
            Posi ="Employee is a Designer.";
        }
        return "Employee Details:"+"\n"+"Employee Number: "+this.emNo+"\n"+"Employee Name:
"+this.emName+"\n"+"Employee Salary: "+this.emSalary+"\n"+"\n"+Posi;
    }


}
```

Program 3: RectangleELC

Field	Value
Program No.	3
Program Name	RectangleELC
Description	Create a BLC class Rectangle

Field	Value
	Attributes : names datatypes ------ ---------- width double-privatre height double-private Implement a parameterized constructor to initialize the non static Field width and height. Methods : --------- 1) Method Name : getArea() Parameter : No Parameters Return Type : double Access modifier : public In this method returns the area of the rectangle. 2) Method Name : getPerimeter() Parameter : No Parameters Return Type : double Access modifier : public In this method returns the perimeter of the rectangle. Note : Don't use toString() method Take one Main class ELC class which is having main method, Create a Rectangle object with width 5 and height 10 and call the getArea() and getPerimeter() methods on it.
Constraints	
Sample Input	5 10

Sample Output	Area = 50.0
	Perimeter = 30.0

Explanation	-
Language(s)	Java
Total Test Cases	3
Test Cases Passed	3
Test Cases Failed	0
Success Rate	100.0%
Score (0–10)	10
Attempts	0
Submitted At	15/11/2025, 11:51:46

Code

```
import java.util.*;
public class Main {
    public static void main(String [] args){
        Scanner sc = new Scanner(System.in);
double width =Double.parseDouble(sc.next());
double height = Double.parseDouble(sc.next());
if(width<=0||height<=0){
    System.out.println("Invalid input! Width and height must be positive numbers.");
    System.exit(0);
}
Rectangle r = new Rectangle(width,height);
 System.out.println("Area of Rectangle: "+r.getArea());
 System.out.println("Perimeter of Rectangle: "+r.getPerimeter());
    }
}
class Rectangle{
    private double width;
    private double height;
    Rectangle(double width,double height){
        this.width = width;
        this.height=height;
    }
    public double getArea(){
        return this.width*this.height;
    }
    public double getPerimeter(){
    return (width+height)*2;
    }
}
```