JAVA ASSIGNMENT 1

QUESTION 1:

INPUT AND OUTPUT:

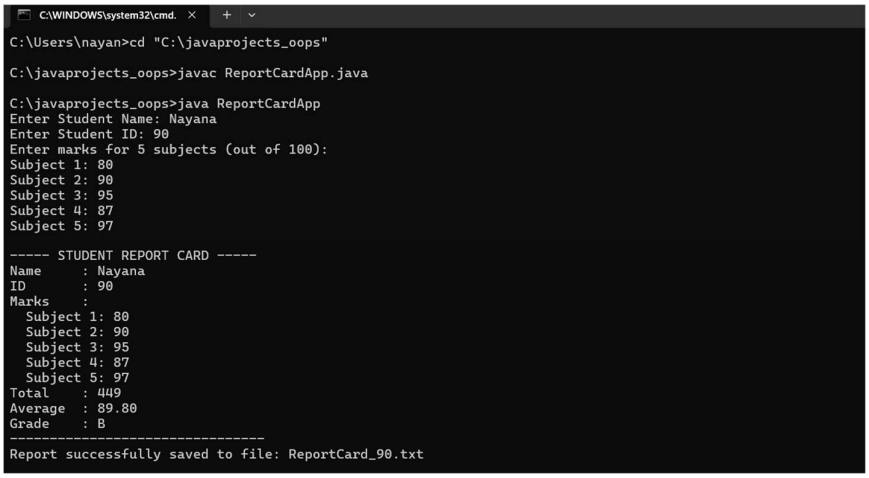
Program 1:

```
import java.util.Scanner;
import java.io.FileWriter;
import java.io.IOException;

class Student {
    String name;
    String id;
    int[] marks = new int[5];
    int total = 0;
    float average;
    char grade;

    void inputDetails(Scanner sc) {
        System.out.print("Enter Student Name: ");
        name = sc.nextLine();

        System.out.print("Enter Student ID: ");
        id = sc.nextLine();

        System.out.println("Enter marks for 5 subjects (out of 100):");
        for (int i = 0; i < 5; i++) {
            while (true) {
                System.out.print("Subject " + (i + 1) + ": ");
                marks[i] = sc.nextInt();
                if (marks[i] >= 0 && marks[i] <= 100) {
                    break;
                } else {
                    System.out.println("Invalid mark! Enter a value between 0 and 100.");
```

```
            }
        }
        total += marks[i];
    }
    average = total / 5.0f;
    assignGrade();
}

void assignGrade() {
    if (average >= 90) grade = 'A';
    else if (average >= 75) grade = 'B';
    else if (average >= 60) grade = 'C';
    else if (average >= 40) grade = 'D';
    else grade = 'F';
}

void displayReport() {
    System.out.println("\n----- STUDENT REPORT CARD -----");
    System.out.println("Name     : " + name);
    System.out.println("ID       : " + id);
    System.out.println("Marks    : ");
    for (int i = 0; i < 5; i++) {
        System.out.println("  Subject " + (i + 1) + ": " + marks[i]);
    }
    System.out.println("Total    : " + total);
    System.out.printf("Average  : %.2f\n", average);
    System.out.println("Grade    : " + grade);
    System.out.println("--------------------------------");
}

void saveToFile() {
    try {
```

```
            FileWriter fw = new FileWriter("ReportCard_" + id + ".txt");
            fw.write("----- STUDENT REPORT CARD -----\n");
            fw.write("Name     : " + name + "\n");
            fw.write("ID       : " + id + "\n");
            fw.write("Marks    :\n");
            for (int i = 0; i < 5; i++) {
                fw.write("  Subject " + (i + 1) + ": " + marks[i] + "\n");
            }
            fw.write("Total    : " + total + "\n");
            fw.write(String.format("Average  : %.2f\n", average));
            fw.write("Grade    : " + grade + "\n");
            fw.write("--------------------------------\n");
            fw.close();
            System.out.println("Report successfully saved to file: ReportCard_" + id + ".txt");
        } catch (IOException e) {
            System.out.println("An error occurred while saving the report: " + e.getMessage());
        }
    }
}

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

        Student student = new Student();
        student.inputDetails(sc);
        student.displayReport();
        student.saveToFile();

        sc.close();
    }
}
```

Output:


C:\WINDOWS\system32\cmd.
C:\Users\nayan>cd "C:\javaprojects_oops"
C:\javaprojects_oops>javac ReportCardApp.java
C:\javaprojects_oops>java ReportCardApp
Enter Student Name: Nayana
Enter Student ID: 90
Enter marks for 5 subjects (out of 100):
Subject 1: 80
Subject 2: 90
Subject 3: 95
Subject 4: 87
Subject 5: 97
----- STUDENT REPORT CARD -----
Name : Nayana
ID : 90
Marks :
Subject 1: 80
Subject 2: 90
Subject 3: 95
Subject 4: 87
Subject 5: 97
Total : 449
Average : 89.80
Grade : B
Report successfully saved to file: ReportCard_90.txt

Program 2:

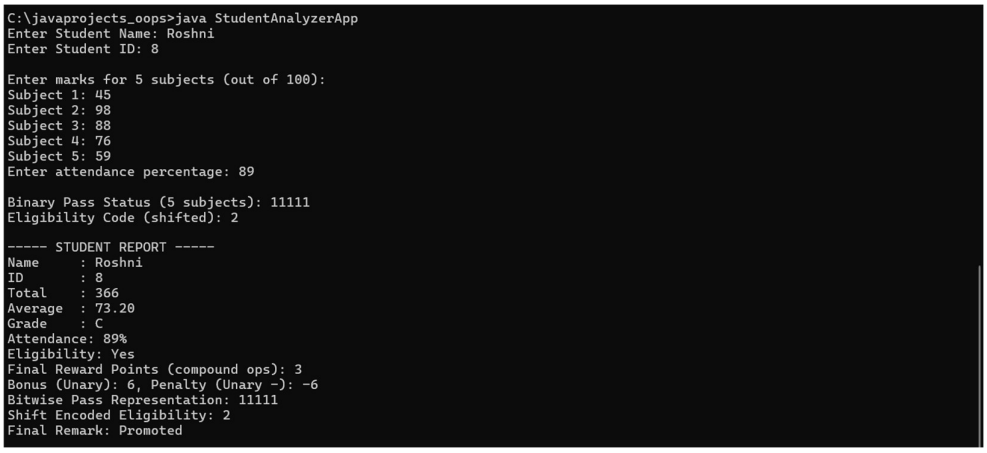
DATA OPERATORS

```
import java.util.Scanner;

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

        // INPUT
        System.out.print("Enter Student Name: ");
        String name = sc.nextLine();
        System.out.print("Enter Student ID: ");
        int id = sc.nextInt();

        int[] marks = new int[5];
        int total = 0;
        int subjectPassBinary = 0;

        System.out.println("\nEnter marks for 5 subjects (out of 100):");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            marks[i] = sc.nextInt();
            total += marks[i];

            // Bitwise: If subject mark >= 40, set the corresponding bit to 1
            if (marks[i] >= 40) {
                subjectPassBinary |= (1 << i);  // Bitwise OR and Shift
            }
        }

        // Arithmetic Operators
        double average = total / 5.0;
```

```
        // Relational + Logical + Ternary
        char grade = (average >= 90) ? 'A' :
                     (average >= 75) ? 'B' :
                     (average >= 60) ? 'C' : 'F';

        System.out.print("Enter attendance percentage: ");
        int attendance = sc.nextInt();
        boolean isEligible = average >= 50 && attendance >= 75;

        // Compound Assignment + Unary
        int reward = 100;
        reward += 20;  // bonus
        reward -= 10;  // minor penalty
        reward *= 2;   // doubled due to participation
        reward /= 3;
        reward %= 7;

        // Unary Increment and Decrement
        int bonus = 5;
        bonus++;       // Post increment
        ++bonus;       // Pre increment
        bonus--;       // Post decrement
        int penalty = -bonus; // Unary minus

        // Bitwise Representation
        System.out.println("\nBinary Pass Status (5 subjects): " + String.format("%5s",
Integer.toBinaryString(subjectPassBinary)).replace(' ', '0'));

        // Shift Operators: Encoding eligibility
        int encoded = isEligible ? (1 << 1) : 0; // Left shift to encode eligibility
        System.out.println("Eligibility Code (shifted): " + encoded);
```

```
        // OUTPUT
        System.out.println("\n----- STUDENT REPORT -----");
        System.out.println("Name     : " + name);
        System.out.println("ID       : " + id);
        System.out.println("Total    : " + total);
        System.out.printf("Average  : %.2f\n", average);
        System.out.println("Grade    : " + grade);
        System.out.println("Attendance: " + attendance + "%");
        System.out.println("Eligibility: " + (isEligible ? "Yes" : "No"));
        System.out.println("Final Reward Points (compound ops): " + reward);
        System.out.println("Bonus (Unary): " + bonus + ", Penalty (Unary -): " + penalty);
        System.out.println("Bitwise Pass Representation: " +
Integer.toBinaryString(subjectPassBinary));
        System.out.println("Shift Encoded Eligibility: " + encoded);
        System.out.println("Final Remark: " + (grade != 'F' && isEligible ? "Promoted" : "Needs
Improvement"));


        sc.close();
    }
}
```

Output:



Program 3:

All datatypes

```
import java.util.Arrays;

public class SmartFarmMonitor {

    public static void main(String[] args) {

        // 1. Primitive Data Types
        byte sensorId = 5;                    // byte
        short fieldZone = 102;                // short
        int totalReadings = 1200;             // int
        long timestamp = System.currentTimeMillis(); // long
        float soilMoisture = 22.5f;           // float (%)
        double temperature = 35.742;          // double (°C)
        char zoneType = 'A';                  // char (crop type)
        boolean isActive = true;              // boolean

        // 2. Bitwise Flags: status flags for alerting
        int status = 0b0000;                  // binary: 4 flags
        final int DRY = 0b0001;               // dry soil
        final int HOT = 0b0010;               // high temperature
        final int SENSOR_ERROR = 0b0100;      // faulty sensor
        final int IRRIGATION_ON = 0b1000;     // irrigation running

        // Logic to raise flags
        if (soilMoisture < 30.0f) status |= DRY;
        if (temperature > 40.0) status |= HOT;
        if (!isActive) status |= SENSOR_ERROR;

        // 3. Non-Primitive Data Types
        String[] sensors = {"Moisture", "Temp", "Humidity", "pH"};
```

```
int[] readings = {23, 41, 68, 6};          // Array
String zoneName = new String("Zone Alpha");

// 4. Wrapper Classes
Integer readingCount = Integer.valueOf(totalReadings);
Double avgTemp = Double.valueOf(temperature);

// Output
System.out.println("=== SMART FARM SENSOR DASHBOARD ===");
System.out.println("Sensor ID        : " + sensorId);
System.out.println("Zone             : " + zoneName + " (" + zoneType + ")");
System.out.println("Field Zone       : " + fieldZone);
System.out.println("Timestamp        : " + timestamp);
System.out.println("Readings Count   : " + readingCount);
System.out.println("Soil Moisture(%) : " + soilMoisture);
System.out.println("Temperature (°C) : " + avgTemp);
System.out.println("Sensors Available: " + Arrays.toString(sensors));

// 5. Logical Decisions
if ((status & DRY) != 0) {
    System.out.println(" ⚠️ ALERT: Soil is too dry!");
}
if ((status & HOT) != 0) {
    System.out.println(" ⚠️ ALERT: High temperature detected!");
}
if ((status & SENSOR_ERROR) != 0) {
    System.out.println(" ❌ ERROR: Sensor malfunction!");
}

// 6. Turn irrigation ON if needed
if ((status & DRY) != 0 && (status & HOT) == 0) {
    status |= IRRIGATION_ON;
```

```
            System.out.println(" 💧 Irrigation turned ON.");
        } else {
            System.out.println(" 💧 Irrigation not needed.");
        }

        // 7. Final Status Report
        System.out.println("Status Binary     : " + String.format("%4s",
Integer.toBinaryString(status)).replace(' ', '0'));
        System.out.println("System Active     : " + isActive);
    }
}
```

Output:

```
C:\javaprojects_oops>java SmartFarmMonitor
=== SMART FARM SENSOR DASHBOARD ===
Sensor ID         : 5
Zone              : Zone Alpha (A)
Field Zone        : 102
Timestamp         : 1753198939303
Readings Count    : 1200
Soil Moisture(%) : 22.5
Temperature (°C) : 35.742
Sensors Available: [Moisture, Temp, Humidity, pH]
?? ALERT: Soil is too dry!
? Irrigation turned ON.
Status Binary     : 1001
System Active     : true
```

Program 4:

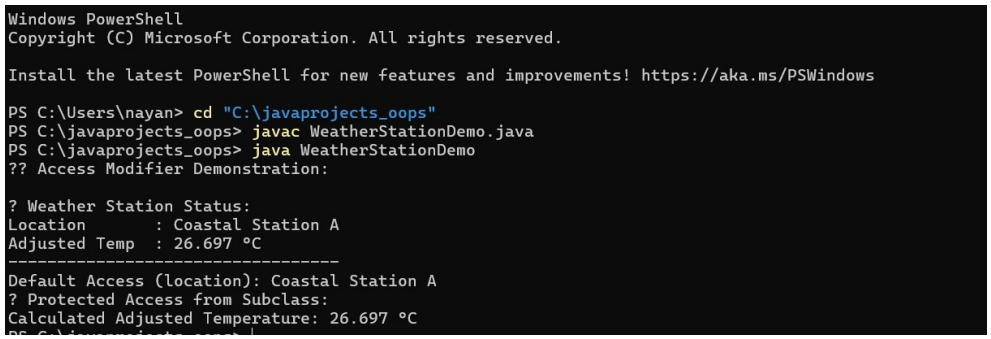
Access modifiers

```
public class WeatherStationDemo {

    public static void main(String[] args) {
        WeatherStation ws = new WeatherStation();
        System.out.println("🛰️ Access Modifier Demonstration:\n");

        // ✅ public method - accessible
        ws.displayStatus();

        // ✅ default field (same class)
        System.out.println("Default Access (location): " + ws.location);

        // ❌ private fields - not accessible
        // System.out.println(ws.temperatureRaw); // ERROR

        // ✅ protected method accessible via subclass
        AdvancedStation adv = new AdvancedStation();
        adv.calculateAndDisplayTemp();
    }
}

// ====================== 🌩️ Weather Station ======================
class WeatherStation {

    // 🔒 PRIVATE: Raw sensor values (only accessible inside this class)
    private double temperatureRaw = 27.345;
    private double humidityRaw = 64.8;
```

```
    // 🏷️ DEFAULT (no modifier): Accessible within same file/package
    String location = "Coastal Station A";

    // 🛡️ PROTECTED: Accessible in subclass or same package
    protected double computeAdjustedTemp() {
        // Simulate adjusting based on humidity
        return temperatureRaw - (humidityRaw * 0.01);
    }

    // 🌐 PUBLIC: Accessible anywhere
    public void displayStatus() {
        System.out.println("📡 Weather Station Status:");
        System.out.println("Location       : " + location);
        System.out.println("Adjusted Temp  : " + computeAdjustedTemp() + " °C");
        System.out.println("---------------------------------");
    }
}

// ====================== 📊 Subclass to Access Protected ======================
class AdvancedStation extends WeatherStation {

    public void calculateAndDisplayTemp() {
        double adjusted = computeAdjustedTemp();  // ✅ protected method
        System.out.println("✅ Protected Access from Subclass:");
        System.out.println("Calculated Adjusted Temperature: " + adjusted + " °C");
    }
}
```

	Default	Private	Protected	Public
Same Class	Yes	Yes	Yes	Yes
Same Package Subclass	Yes	No	Yes	Yes
Same Package Non-Subclass	Yes	No	Yes	Yes
Different Package Subclass	No	No	Yes	Yes
Different Package Non-Subclass	No	No	No	Yes

Output:



Program 5:

CONTROL STATEMENTS

```
import java.util.Scanner;

public class SmartHomeEnergyOptimizer {

    static final int MAX_DEVICES = 5;

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String[] devices = {"Fan", "Light", "AC", "TV", "Heater"};
        int[] powerRatings = {70, 40, 150, 100, 200}; // watts
```

```
boolean[] isOn = new boolean[MAX_DEVICES];

System.out.println("👋 Welcome to Smart Home Energy Optimizer");

// DO-WHILE: Accepting usage time
int hour;
do {
    System.out.print("Enter current hour (0–23): ");
    hour = sc.nextInt();
} while (hour < 0 || hour > 23);

// SWITCH: Time-based automation
switch (hour) {
    case 6: case 7: case 8:
        System.out.println("🌅 Morning Mode: Turning on Fan and Light");
        isOn[0] = true; // Fan
        isOn[1] = true; // Light
        break;
    case 22: case 23: case 0:
        System.out.println("🌙 Night Mode: Only Fan remains on");
        isOn[0] = true;
        break;
    default:
        System.out.println("🕒 Day Mode: Manual Control");
        break;
}

// FOR loop: Manual toggle
for (int i = 0; i < MAX_DEVICES; i++) {
    System.out.print("Turn ON " + devices[i] + "? (yes/no): ");
    String input = sc.next().toLowerCase();
```

```
    if (input.equals("skip")) {
        continue; // skip this device
    }

    if (input.equals("exit")) {
        System.out.println("Exiting setup...");
        break; // jump out of loop
    }

    isOn[i] = input.equals("yes");
}

// WHILE: Power calculation
int index = 0;
int totalPower = 0;
while (index < MAX_DEVICES) {
    if (isOn[index]) {
        totalPower += powerRatings[index];
    }
    index++;
}

System.out.println("\n📊 Total Power Consumed: " + totalPower + " watts");

// IF-ELSE: Power optimization warning
if (totalPower > 400) {
    System.out.println("⚠️ High power usage! Consider turning off some devices.");
} else {
    System.out.println("✅ Power usage is optimal.");
}
```

```
        // Bitwise Example
        System.out.println("\n🔍 Bitwise Check:");
        int statusBits = 0;
        for (int i = 0; i < MAX_DEVICES; i++) {
            if (isOn[i]) {
                statusBits |= (1 << i); // set ith bit
            }
        }
        System.out.println("Device Status Bitmask: " + Integer.toBinaryString(statusBits));

        // RETURN to exit
        System.out.println("👋 System shutting down...");
        return;
    }
}
```

Output:

```
PS C:\javaprojects_oops> javac SmartHomeEnergyOptimizer.java
PS C:\javaprojects_oops> java SmartHomeEnergyOptimizer
? Welcome to Smart Home Energy Optimizer
Enter current hour (0?23): 8
? Morning Mode: Turning on Fan and Light
Turn ON Fan? (yes/no): yes
Turn ON Light? (yes/no): yes
Turn ON AC? (yes/no): no
Turn ON TV? (yes/no): no
Turn ON Heater? (yes/no): yes

? Total Power Consumed: 310 watts
? Power usage is optimal.

? Bitwise Check:
Device Status Bitmask: 10011
? System shutting down...
PS C:\javaprojects_oops>
```