

ASSIGNMENT-6.4

2303A51595

B-10

TASK – 1

Prompt :

Write a Java program to accept day-wise product sales using Scanner and calculate daily totals and overall monthly sales.

Display a summary report and allow the user to repeat the process for another month using loops.

Code :

```
import java.util.Scanner;
```

```
class Student {  
    String name;  
    int rollNumber;  
    int marks;  
    Student(String name, int rollNumber, int marks) {  
        this.name = name;  
        this.rollNumber = rollNumber;  
        this.marks = marks;  
    }  
  
    void displayDetails() {  
        System.out.println("Name: " + name);  
        System.out.println("Roll Number: " + rollNumber);  
        System.out.println("Marks: " + marks);  
    }  
  
    String checkPerformance(int average) {  
        if (marks > average)  
            return "Above Class Average";  
        else  
            return "Below Class Average";  
    }  
}  
  
public class StudentTest {
```

```

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    System.out.print("Enter Name: ");
    String name = sc.nextLine();

    System.out.print("Enter Roll Number: ");
    int roll = sc.nextInt();

    System.out.print("Enter Marks: ");
    int marks = sc.nextInt();

    System.out.print("Enter Class Average: ");
    int avg = sc.nextInt();

    Student s = new Student(name, roll, marks);

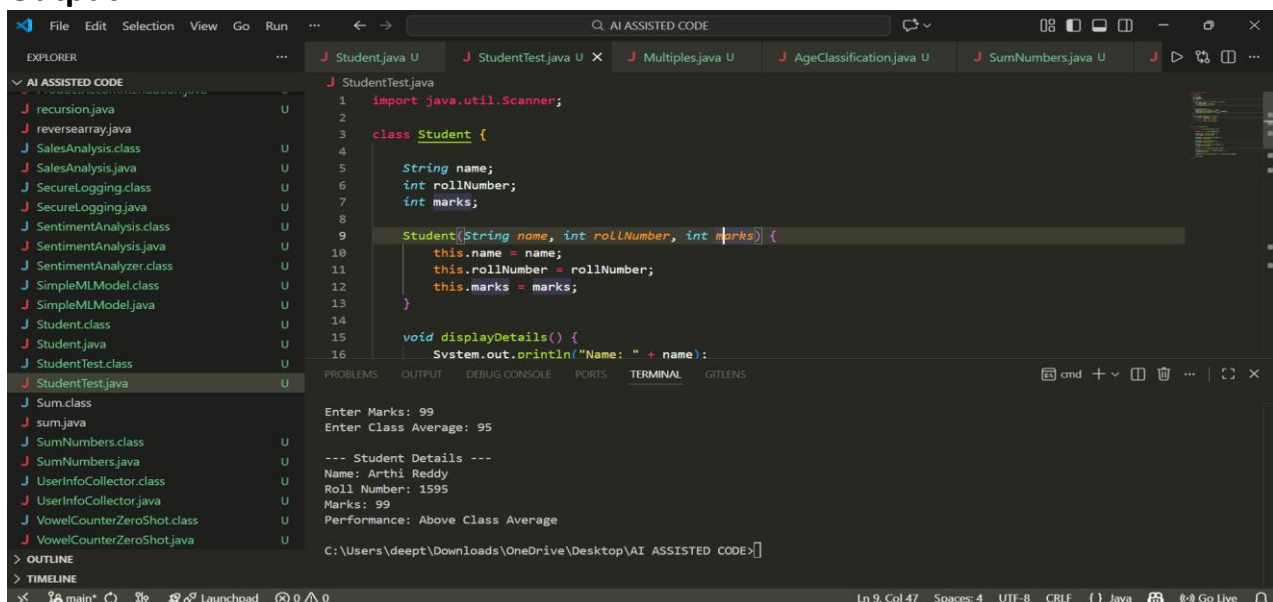
    System.out.println("\n--- Student Details ---");
    s.displayDetails();

    System.out.println("Performance: " + s.checkPerformance(avg));

    sc.close();
}

```

Output :



The screenshot shows an IDE with the following components:

- EXPLORER:** A list of files on the left, including `recursion.java`, `reversearray.java`, `SalesAnalysis.class`, `SecureLogging.class`, `SentimentAnalysis.class`, `SimpleMLModel.class`, `Student.class`, `StudentTest.class`, `Sum.class`, `sum.java`, `SumNumbers.class`, `UserInfoCollector.class`, `VowelCounterZeroShot.class`, and `VowelCounterZeroShot.java`.
- EDITOR:** The `StudentTest.java` file is open, showing the following code:


```

1  import java.util.Scanner;
2
3  class Student {
4
5      String name;
6      int rollNumber;
7      int marks;
8
9      Student(String name, int rollNumber, int marks) {
10         this.name = name;
11         this.rollNumber = rollNumber;
12         this.marks = marks;
13     }
14
15     void displayDetails() {
16         System.out.println("Name: " + name);

```
- TERMINAL:** The output of the program is displayed:


```

Enter Marks: 99
Enter Class Average: 95

--- Student Details ---
Name: Arthi Reddy
Roll Number: 1595
Marks: 99
Performance: Above Class Average

```
- STATUS BAR:** At the bottom, it shows "Ln 9, Col 47", "Spaces: 4", "UTF-8", "CRLF", and "Java".

Analysis :

This program takes product sales details for each day and calculates daily and monthly totals using loops.

It shows the final sales summary and allows the user to repeat the process for the next month.

TASK-2**Prompt :**

Write a Java program to loop through sensor readings, identify even numbers, calculate their square, and print the result clearly.

Code :

```
import java.util.*;

public class SensorMonitoring {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of readings: ");

        int n = sc.nextInt();

        int[] readings = new int[n];

        System.out.println("Enter sensor readings:");

        for(int i = 0; i < n; i++) {

            readings[i] = sc.nextInt;

            // Copilot: check each reading, if even calculate square and print nicely

            for(int value : readings) {

                if(value % 2 == 0) {

                    int square = value * value;

                    System.out.println("Even: " + value + " -> Square: " + square)

                }

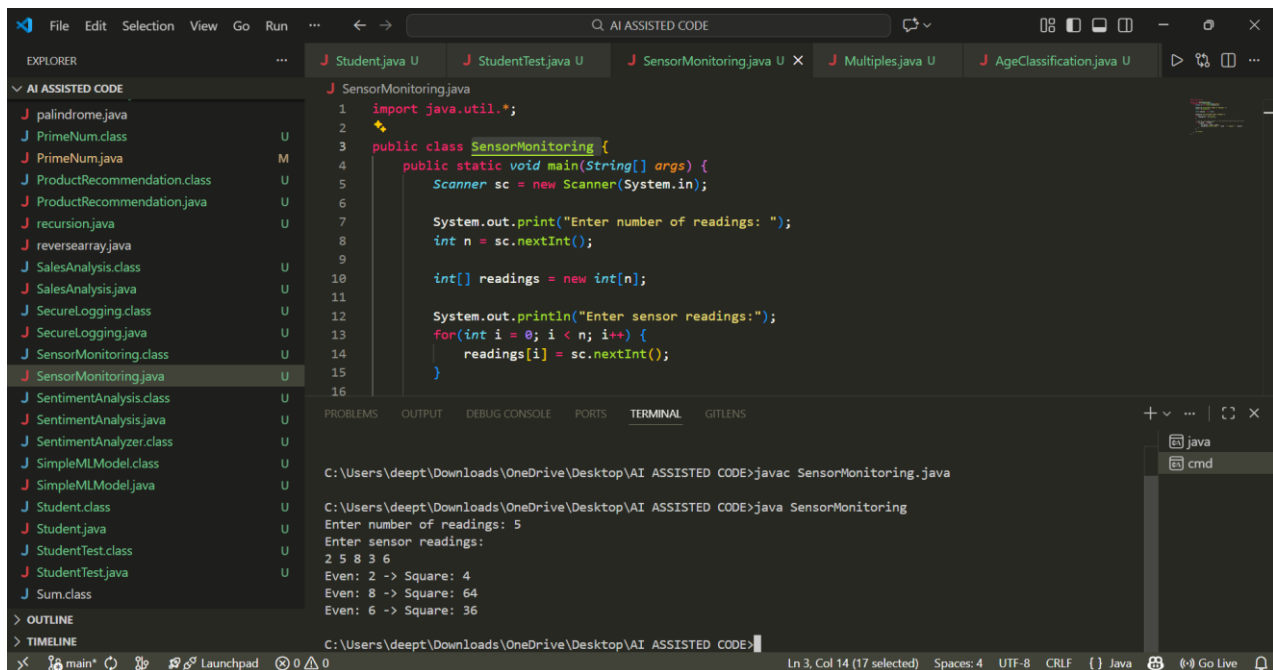
            }

        }

    }

}
```

Output :



The screenshot shows an IDE with the following components:

- EXPLORER:** A list of files including `palindrome.java`, `PrimeNum.class`, `PrimeNum.java`, `ProductRecommendation.class`, `ProductRecommendation.java`, `recursion.java`, `reversearray.java`, `SalesAnalysis.class`, `SalesAnalysis.java`, `SecureLogging.class`, `SecureLogging.java`, `SensorMonitoring.class`, `SensorMonitoring.java`, `SentimentAnalysis.class`, `SentimentAnalysis.java`, `SentimentAnalyzer.class`, `SimpleMLModel.class`, `SimpleMLModel.java`, `Student.class`, `Student.java`, `StudentTest.class`, `StudentTest.java`, and `Sum.class`.
- EDITOR:** The `SensorMonitoring.java` file is open, showing the following code:

```
1 import java.util.*;
2
3 public class SensorMonitoring {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Enter number of readings: ");
8         int n = sc.nextInt();
9
10        int[] readings = new int[n];
11
12        System.out.println("Enter sensor readings:");
13        for(int i = 0; i < n; i++) {
14            readings[i] = sc.nextInt();
15        }
16    }
17 }
```
- TERMINAL:** The output of the program is shown:

```
C:\Users\deept\Downloads\OneDrive\Desktop\AI ASSISTED CODE>javac SensorMonitoring.java
C:\Users\deept\Downloads\OneDrive\Desktop\AI ASSISTED CODE>java SensorMonitoring
Enter number of readings: 5
Enter sensor readings:
2 5 8 3 6
Even: 2 -> Square: 4
Even: 8 -> Square: 64
Even: 6 -> Square: 36
```

Analysis :

The loop checks each reading using `% 2 == 0` to find even numbers. If even, it calculates the square and prints the result clearly.

TASK-3

Prompt :

Write a Java class `BankAccount` with `accountHolder` and `balance`. Add methods for `deposit`, `withdraw` (with insufficient balance check), and `display balance` using if-else conditions.

Code :

```
import java.util.Scanner;
class BankAccount {
    String accountHolder;
    double balance;
    BankAccount(String accountHolder, double balance) {
        this.accountHolder = accountHolder;
        this.balance = balance;
    }
    void deposit(double amount) {
        balance += amount;
    }
}
```

```

        System.out.println("Deposited: " + amount);
    }
    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance! Withdrawal failed.");
        }
    }
}

public class BankSimulation {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter account holder name: ");
        String name = sc.nextLine();
        System.out.print("Enter initial balance: ");
        double bal = sc.nextDouble();
        BankAccount acc = new BankAccount(name, bal);
        System.out.print("Enter deposit amount: ");
        acc.deposit(sc.nextDouble());
        System.out.print("Enter withdrawal amount: ");
        acc.withdraw(sc.nextDouble());
        acc.checkBalance();
        sc.close();
    }
}

```

Output :

The screenshot shows an IDE with the following components:

- EXPLORER:** A list of files on the left, including `BankSimulation.java` which is currently selected.
- EDITOR:** The main area showing the source code of `BankSimulation.java`. The code defines a `BankAccount` class with `deposit`, `withdraw`, and `checkBalance` methods, and a `BankSimulation` class with a `main` method that uses a `Scanner` to take user input.
- TERMINAL:** The bottom panel shows the command `javac BankSimulation.java` and the execution of `java BankSimulation`. The output shows the program running with the following user inputs and results:
 - Enter account holder name: arthi reddy
 - Enter initial balance: 5000
 - Enter deposit amount: 1000
 - Deposited: 1000.0
 - Enter withdrawal amount: 7000
 - Insufficient balance! Withdrawal failed.
 - Current Balance: 6000.0

Analysis :

The class stores account details and updates balance using deposit and withdraw methods. If withdrawal is more than balance, an if-else check prevents it and shows a friendly message.

Task-4**Prompt :**

Write a Java program that stores student names and scores, use a while loop to check each student, and print names of students scoring more than 75.

Code :

```
import java.util.Scanner;

public class ScholarshipCheck {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of students: ");
        int n = sc.nextInt();
        sc.nextLine();

        String[] names = new String[n];
        int[] scores = new int[n];

        // initialize student data
        for (int i = 0; i < n; i++) {
            System.out.print("Enter name: ");
            names[i] = sc.nextLine();

            System.out.print("Enter score: ");
            scores[i] = sc.nextInt();
            sc.nextLine();
        }

        int i = 0;
        System.out.println("\nEligible Students:");
        while (i < n) {
```

```

        if (scores[i] > 75) {
            System.out.println(names[i]);
        }
        i++;
    }
    sc.close();
}
}

```

Output :

```

1  import java.util.Scanner;
2
3  public class ScholarshipCheck {
4      public static void main(String[] args) {

```

Microsoft Windows [Version 10.0.26200.7705]
(c) Microsoft Corporation. All rights reserved.

C:\Users\deept\Downloads\OneDrive\Desktop\AI ASSISTED CODE>javac ScholarshipCheck.java

C:\Users\deept\Downloads\OneDrive\Desktop\AI ASSISTED CODE>java ScholarshipCheck

Enter number of students: 5

Enter name: Arthi reddy

Enter score: 90

Enter name: Varsha

Enter score: 95

Enter name: Rashmitha

Enter score: 90

Enter name: Madhuri

Enter score: 98

Enter name: Nithya

Enter score: 95

Eligible Students:

Arthi reddy

Varsha

Rashmitha

Madhuri

Nithya

Analysis :

The while loop checks each student one by one using an index.

If score is greater than 75, the student name is printed as eligible.

TASK-5

Prompt :

Write a Java class ShoppingCart with an item list. Add methods to add items, remove items, calculate total bill using loops, and apply discount if total exceeds a limit.

Code :

```

import java.util.*;
class Item {
    String name;
    double price;

```

```

int quantity;

Item(String name, double price, int quantity) {
    this.name = name;
    this.price = price;
    this.quantity = quantity;
}
}
class ShoppingCart {
    ArrayList<Item> items = new ArrayList<>();

    // add item to cart
    void addItem(String name, double price, int quantity) {
        items.add(new Item(name, price, quantity));
        System.out.println("Item added to cart.");
    }
    // remove item from cart
    void removeItem(String name) {
        items.removeIf(item -> item.name.equalsIgnoreCase(name));
        System.out.println("Item removed from cart.");
    }
    // calculate total with discount
    void calculateBill() {
        double total = 0;

        for (Item item : items) {
            total += item.price * item.quantity;
        }

        System.out.println("Total before discount: " + total);

        if (total > 1000) {
            total = total * 0.9; // 10% discount
            System.out.println("Discount applied (10%)");
        }
        System.out.println("Final Bill: " + total);
    }
}
public class ShoppingCartDemo {
    public static void main(String[] args) {

```



```

Scanner sc = new Scanner(System.in);
ShoppingCart cart = new ShoppingCart();
System.out.print("Enter number of items: ");
int n = sc.nextInt();
sc.nextLine();

for (int i = 0; i < n; i++) {
    System.out.print("Item name: ");
    String name = sc.nextLine();

    System.out.print("Price: ");
    double price = sc.nextDouble();

    cart.addItem(name, price, qty);
}

System.out.print("Enter item name to remove: ");
cart.removeItem(sc.nextLine());
cart.calculateBill();
sc.close();
}
}

```

Output :

```

C:\Users\deept\Downloads\OneDrive\Desktop\AI ASSISTED CODE>java ShoppingCartDemo
Enter number of items: 2
Item name: rice
Price: 50
Quantity: 1
Item added to cart.
Item name: soap
Price: 40
Quantity: 3
Item added to cart.
Enter item name to remove: rice
Item removed from cart.
Total before discount: 120.0
Final Bill: 120.0

```

Analysis :

Items are stored in an ArrayList and processed using loops to calculate the total. If the bill exceeds the limit, an if-condition applies a discount automatically.