

## Assignment-6.3

2303a51595

B-10

### TASK-1

#### Prompt :

Write a Java Student class with attributes name, roll number, and branch.  
Include constructor and displayDetails() method.

#### Code :

```
import java.util.Scanner;

class Student {
    String name;
    int rollNo;
    String branch;
    Student(String name, int rollNo, String branch) {
        this.name = name;
        this.rollNo = rollNo;
        this.branch = branch;
    }
    void displayDetails() {
        System.out.println("\n--- Student Details ---");
        System.out.println("Name: " + name);
        System.out.println("Roll No: " + rollNo);
        System.out.println("Branch: " + branch);
    }
    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 rollNo = sc.nextInt();
sc.nextLine(); // clear buffer

System.out.print("Enter Branch: ");
String branch = sc.nextLine();

Student s1 = new Student(name, rollNo, branch);

s1.displayDetails();

sc.close();
}
}

```

### Output :

The screenshot shows a Java development environment with multiple tabs open at the top: PrimeNum.java, Student.java (selected), EnergyBillCalculator.java, fibo.java, and CentimeterConverter.java. The Explorer sidebar on the left lists various Java files. The Student.java code is displayed in the main editor:

```

import java.util.Scanner;

class Student {
    String name;
    int rollNo;
    String branch;
    Student(String name, int rollNo, String branch) {
        this.name = name;
        this.rollNo = rollNo;
        this.branch = branch;
    }
    void displayDetails() {
        System.out.println("\n--- Student Details ---");
        System.out.println("Name: " + name);
        System.out.println("Roll No: " + rollNo);
        System.out.println("Branch: " + branch);
    }
    public static void main(String[] args) {
}

```

The terminal window below shows the execution of the program:

```

C:\Users\deeps\Downloads\OneDrive\Desktop\AI ASSISTED CODE>java Student
Enter Name: Arthi Reddy
Enter Roll Number: 1595
Enter Branch: CSE
--- Student Details ---
Name: Arthi Reddy
Roll No: 1595
Branch: CSE

```

### Explanation :

This program uses a `Student` class with constructor and methods to store and display student details.

It demonstrates object creation and basic Object-Oriented Programming concepts.

## TASK-2

### Prompt :

Write a Java program to take a number from the user and print its first 10 multiples.

Use both for loop and while loop to show the result.

### Code :

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

```
class Multiples {
```

```
    static void multiplesFor(int n) {
        System.out.println("\nUsing For Loop:");
        for(int i = 1; i <= 10; i++) {
            System.out.println(n * i);
        }
    }

    static void multiplesWhile(int n) {
        System.out.println("\nUsing While Loop:");
        int i = 1;
        while(i <= 10) {
            System.out.println(n * i);
            iString[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = sc.nextInt();

        multiplesFor(n);
        multiplesWhile(n);

        sc.close();
    }
}
```

## **Output :**

The screenshot shows a Java code editor interface with several files listed in the sidebar under 'EXPLORER' and 'AI ASSISTED CODE'. The main editor area displays `Multiples.java`. The code uses a combination of for and while loops to print multiples of a user-specified number. The terminal below shows the output of the program.

```
class Multiples {
    public static void main(String[] args) {
        System.out.print("Enter a number: ");
        int n = sc.nextInt();

        multiplesFor(n);
        multiplesWhile(n);

        sc.close();
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL GITLENS

```
Using For Loop:
3
6
9
12
15
18
21
24
27
30

Using While Loop:
3
6
9
```

## **Explanation :**

This program takes a number from the user and prints its first 10 multiples using both for and while loops

### **TASK-3**

## Prompt :

Write a Java program to take age as input from the user and classify the person as child, teenager, adult, or senior using conditional statements.

## **Code :**

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

```
class AgeClassification {
```

```
static String classifyAge(int age) {
```

```
if(age < 13)
    return "Child";
else if(age < 20)
    return "Teenager";
```

```

        else if(age < 60)
            return "Adult";
        else
            return "Senior";
    }

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

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

    String result = classifyAge(age);

    System.out.println("Category: " + result);

    sc.close();
}
}

```

## Output :

The screenshot shows a Java development environment with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Toolbar:** Standard icons for file operations.
- Explorer:** Shows a list of Java files in the project, including PrimeNum.java, Student.java, Multiples.java, AgeClassification.java, Bank.class, Bill.java, CentimeterConverter.class, CentimeterConverter.java, Duplicate.java, EnergyBillCalculator.class, EnergyBillCalculator.java, EthicalLogging.class, Fibo.java, FileLineCount.class, FileLineCount.java, Java.class, Multiples.class, Multiples.java, NameFormatter.class, NameFormatter.java, Palindrome.java, PrimeNum.class, PrimeNum.java, and Main.java.
- Code Editor:** The code for `AgeClassification.java` is displayed. It contains a class definition with a static method `classifyAge` that returns a string based on the input age. An `else if` condition is highlighted in red.
- Terminal:** Shows the command-line interface where the Java compiler (`javac`) and interpreter (`java`) are used to run the program. The output shows the user entering an age of 19 and the program outputting "Teenager".
- Status Bar:** Shows the current file is `Main.java`, line 11, column 26, with 4 spaces, encoding is UTF-8, CRLF, Java selected, and other standard status indicators.

### **Explanation :**

This program takes age as input and checks conditions using if-else statements to decide the age group.

It demonstrates decision making and conditional logic in Java.

### **TASK-4**

#### **Prompt :**

Write a Java program to take a number n from the user and calculate the sum of the first n natural numbers.

#### **Code :**

```
import java.util.Scanner;

class SumNumbers {
    static int sumFor(int n) {
        int sum = 0;
        for(int i = 1; i <= n; i++) {
            sum += i;
        }
        return sum;
    }
    static int sumWhile(int n) {
        int sum = 0, i = 1;
        while(i <= n) {
            sum += i;
            i++;
        }
        return sum;
    }
    static int sumFormula(int n) {
        return n * (n + 1) / 2;
    }

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

```

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

System.out.println("Sum using For loop: " + sumFor(n));
System.out.println("Sum using While loop: " + sumWhile(n));
System.out.println("Sum using Formula: " + sumFormula(n));

sc.close();
}
}

```

## Output :

The screenshot shows a Java development environment with the following details:

- File Explorer:** Shows multiple Java files in the AI ASSISTED CODE folder, including PrimeNum.java, Student.java, Multiples.java, AgeClassification.java, and SumNumbers.java.
- Code Editor:** Displays the content of SumNumbers.java. The code defines three methods: sumFor, sumWhile, and sumFormula. The sumWhile method is currently selected.
- Terminal:** Shows the command-line output of running the SumNumbers.java file. It prompts for input ("Enter n: 3") and then displays three results: "Sum using For loop: 6", "Sum using While loop: 6", and "Sum using Formula: 6".
- Bottom Status Bar:** Provides information about the current file (C:\Users\deeps\Downloads\OneDrive\Desktop\AI ASSISTED CODE\SumNumbers.java), line count (Ln 13, Col 24), and encoding (UTF-8 CRLF).

## Explanation :

This program calculates the sum of first n numbers using loops and a mathematical formula.

## Task-5

### Prompt :

Write a Java program to create a BankAccount class with deposit, withdraw, and check balance methods.

Code :

```
import java.util.Scanner;

class BankAccount {

    String name;
    double balance;

    // Constructor
    BankAccount(String name, double balance) {
        this.name = name;
        this.balance = balance;
    }

    // Deposit money
    void deposit(double amount) {
        balance += amount;
        System.out.println("Amount Deposited: " + amount);
    }

    // Withdraw money
    void withdraw(double amount) {
        if(amount <= balance) {
            balance -= amount;
            System.out.println("Amount Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient Balance");
        }
    }

    // Check balance
    void checkBalance() {
        System.out.println("Current Balance: " + balance);
    }
}

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

```

Scanner sc = new Scanner(System.in);

System.out.print("Enter Account Holder Name: ");
String name = sc.nextLine();
BankAccount acc = new BankAccount(name, balance);

System.out.print("Enter deposit amount: ");
acc.deposit(sc.nextDouble());

System.out.print("Enter withdraw amount: ");
acc.withdraw(sc.nextDouble());

acc.checkBalance();

sc.close();
}
}

```

The screenshot shows a Java development environment with the following details:

- File Explorer:** Shows multiple Java files in the "AI ASSISTED CODE" folder.
- Editor:** Displays the content of `BankAccount.java`.
- Terminal:** Shows the execution of the program and its output.

```

import java.util.Scanner;
class BankAccount {
    String name;
    double balance;
    // Constructor
    BankAccount(String name, double balance) {
        this.name = name;
        this.balance = balance;
    }
    // Deposit money
    void deposit(double amount) {
        balance += amount;
        System.out.println("Amount Deposited: " + amount);
    }
    // Withdraw money
    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Amount Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient Balance");
        }
    }
    // Check Balance
    void checkBalance() {
        System.out.println("Current Balance: " + balance);
    }
}

```

**Terminal Output:**

```

C:\Users\deeps\Downloads\OneDrive\Desktop\AI ASSISTED CODE>java BankAccount
Enter Account Holder Name: Arathi reddy
Enter Initial Balance: 5000
Enter deposit amount: 3000
Amount Deposited: 3000.0
Enter withdraw amount: 2000
Amount Withdrawn: 2000.0
Current Balance: 6000.0

```

## Explanation :

This program creates a `BankAccount` class to manage money transactions like deposit, withdraw, and balance checking.

