Object Orientated Programming II Assignment.

Name: Philomen Khamal Urendi

Reg No: SCT221-0799/2022

SECTION A:

1. Explain the differences between primitive and reference data types.

Primitive Data Types are the most basic data types available within the Java language. They are predefined by the language and named by a keyword. Examples: int, byte, short, long, float, double, char and Boolean.

Reference Data Types are any objects or arrays created in the program. They are created using defined constructors of the classes. Examples: Instances of classes (String, ArrayList, etc.), arrays.

2. Define the scope of a variable (hint: local and global variable)

Local Variables are declared within a method or a block of code. They are only accessible within the scope they are defined in.

Global Variables: Variables declared outside of any method, usually at the class level. They are accessible throughout the entire class.

3. Why is initialization of variables required.

Initializing variables ensures that they have a valid value before they are used, which helps prevent runtime errors and unexpected behavior.

4. Differentiate between static, instance and local variables.

Static Variables: These are variables that belong to the class itself, not to any specific instance of the class. They are shared among all instances of the class and can be accessed without creating an object.

Instance Variables: These are variables that belong to a specific instance of a class. Each object has its own copy of the instance variables.

Local Variables: These are variables that are defined within a method or a block of code. They are only accessible within the scope they are defined in.

5. Differentiate between widening and narrowing casting in java.

Widening Casting: This is the automatic conversion of a smaller data type to a larger data type, such as converting an int to a double. Widening casting is safe and does not result in a loss of data.

Narrowing Casting: This is the manual conversion of a larger data type to a smaller data type, such as converting a double to an int. Narrowing casting can result in a loss of data or precision and should be done with caution.

6. The following table shows data type, its size, default value and the range. Filling in the missing values.

TYPE	SIZE (IN BYTES)	DEFAULT	RANGE
boolean	1 bit	True, False	true, false
Char	2	'\0'	'\0000' to '\ffff'
Byte	1	0	-128 to 127
Short	2	0	-32,768 to 32,767
Int	4	0	-2,147,483,648 to 2,147,483,647
Long	8	0L	- 9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
Float	4	00.0f	-3.4028235E+38 to 3.4028235E+38
Double	8	0.0	-1.8E+308 to +1.8E+308

7. Define class as used in OOP

A class is a blueprint or template for creating objects. It defines a set of attributes and methods that the created objects will have.

8. Explain the importance of classes in Java programming.

Encapsulation: A class encapsulates data (attributes) and the functions (methods) that operate on that data, hiding the internal implementation details from the outside world.

Abstraction: A class provides an abstraction of real-world objects, focusing on their essential characteristics and behaviors, while hiding the unnecessary details.

Inheritance: A class can inherit properties and methods from a parent class, allowing for code reuse and the creation of hierarchical relationships between classes.

Polymorphism: Objects of different classes can be treated as objects of a common superclass, allowing for dynamic binding and method overriding.

SECTION B:

1.

```
import java.util.Scanner;

public class Main {
   public static void main(String[] args) {
      // Create a Scanner object for reading input
      Scanner scanner = new Scanner(System.in);
```

```
        // Prompt the user to enter their surname
        System.out.print("Enter your surname: ");
        String surname = scanner.nextLine();

        // Prompt the user to enter their current age
        System.out.print("Enter your current age: ");
        int age = scanner.nextInt();

        // Calculate the number of characters in the surname
        int surnameLength = surname.length();

        // Determine if the age is even or odd
        String ageType = (age % 2 == 0) ? "even" : "odd";

        // Print the results
        System.out.println("The number of characters in your surname is: " + surnameLength);
        System.out.println("Your current age is an " + ageType + " number.");
    }
}
```

2.

```
import java.util.Scanner;



public class StudentMarkAverage {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);



        // Prompt the user to enter the marks for 5 units

        System.out.println("Enter the marks for 5 units:");



        // Read the marks for each unit

        double unit1, unit2, unit3, unit4, unit5;

        System.out.print("Unit 1: ");

        unit1 = scanner.nextDouble();
```

```
        System.out.print("Unit 2: ");
        unit2 = scanner.nextDouble();
        System.out.print("Unit 3: ");
        unit3 = scanner.nextDouble();
        System.out.print("Unit 4: ");
        unit4 = scanner.nextDouble();
        System.out.print("Unit 5: ");
        unit5 = scanner.nextDouble();

        // Calculate the average
        double average = (unit1 + unit2 + unit3 + unit4 + unit5) / 5;

        // Display the average
        System.out.printf("The average of the 5 units is: %.2f", average);
    }
}
```

3.

```
import java.util.Scanner;

public class DivisibilityChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```
    System.out.print("Enter an integer: ");
    int number = scanner.nextInt();

    System.out.println("Checking divisibility for number: " + number);
    checkDivisibility(number);
}

private static void checkDivisibility(int number) {
    for (int i = 1; i <= 9; i++) {
        if (isDivisible(number, i)) {
            printDivisibilityReason(number, i);
        }
    }
}

private static boolean isDivisible(int number, int divisor) {
    return number % divisor == 0;
}

private static void printDivisibilityReason(int number, int divisor) {
    switch (divisor) {
        case 1:
            System.out.println(number + " is divisible by 1 because any number is divisible by 1.");
            break;
```

```
            case 2:
                if (number % 2 == 0) {
                    System.out.println(number + " is divisible by 2 because it is even.");
                }
                break;
            case 3:
                if (sumOfDigits(number) % 3 == 0) {
                    System.out.println(number + " is divisible by 3 because the sum of its digits is divisible by
3.");
                }
                break;
            case 4:
                if (number % 4 == 0) {
                    System.out.println(number + " is divisible by 4 because the last two digits form a number
divisible by 4.");
                }
                break;
            case 5:
                if (number % 5 == 0) {
                    if (number % 10 == 0) {
                        System.out.println(number + " is divisible by 5 because it ends with a 0.");
                    } else if (number % 10 == 5) {
                        System.out.println(number + " is divisible by 5 because it ends with a 5.");
                    }
                }
```

```
            break;
        case 6:
            if (number % 2 == 0 && number % 3 == 0) {
                System.out.println(number + " is divisible by 6 because it is divisible by both 2 and 3.");
            }
            break;
        case 7:
            // 7 does not have a simple rule, so we just check divisibility
            if (number % 7 == 0) {
                System.out.println(number + " is divisible by 7.");
            }
            break;
        case 8:
            if (number % 8 == 0) {
                System.out.println(number + " is divisible by 8 because the last three digits form a number divisible by 8.");
            }
            break;
        case 9:
            if (sumOfDigits(number) % 9 == 0) {
                System.out.println(number + " is divisible by 9 because the sum of its digits is divisible by 9.");
            }
            break;
        default:
```

```
            break;
        }
    }

    private static int sumOfDigits(int number) {
        int sum = 0;
        while (number != 0) {
            sum += number % 10;
            number /= 10;
        }
        return sum;
    }
}
```

4.

```
public class Multiples {
    public static void main(String[] args) {
        int start = 71;
        int end = 150;

        System.out.println("Multiples of 2, 3, and 7 within the range 71 to 150:");

        for (int i = start; i <= end; i++) {
            if (i % 2 == 0 || i % 3 == 0 || i % 7 == 0) {
                System.out.println(i);
            }
```

```
    }
  }
}
```

5.

```
import java.util.Scanner;

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

    // Prompt user to enter the first number
    System.out.print("Enter the first number: ");
    double firstNumber = scanner.nextDouble();

    // Prompt user to enter the operation
    System.out.print("Enter an operator (+, -, *, /): ");
    char operator = scanner.next().charAt(0);

    // Prompt user to enter the second number
    System.out.print("Enter the second number: ");
    double secondNumber = scanner.nextDouble();

    double result;

    // Perform the operation
    switch (operator) {
      case '+':
        result = firstNumber + secondNumber;
```

```
            break;
        case '-':
            result = firstNumber - secondNumber;
            break;
        case '*':
            result = firstNumber * secondNumber;
            break;
        case '/':
            // Check for division by zero
            if (secondNumber != 0) {
                result = firstNumber / secondNumber;
            } else {
                System.out.println("Error: Division by zero is not allowed.");
                return;
            }
            break;
        default:
            System.out.println("Error: Invalid operator.");
            return;
    }

    // Display the result
    System.out.println("The result is: " + result);
  }
}
```