SCHOOL OF COMPUTER SCIENCE AND IT
YBN UNIVERSITY RANCHI

JAVA PROGRAMMING
ASSIGNMENT 10

Problem 1: Method Overloading for Calculating Areas

Description:
Write a Java program that includes overloaded methods named `calculateArea` to compute the area of different geometric shapes: a circle, a rectangle, and a triangle. The program should include:

- `calculateArea(double radius)` to calculate the area of a circle.
- `calculateArea(double length, double width)` to calculate the area of a rectangle.
- `calculateArea(double base, double height)` to calculate the area of a triangle.

The program should prompt the user to choose the shape and input the relevant dimensions, then call the appropriate overloaded method and print the area.

Concepts Covered:

- Method overloading
- Using method parameters with different signatures

Hint:

- For the triangle, use the formula: `0.5 * base * height`.
- Use method overloading to define multiple methods with the same name but different parameter lists.

Problem 2: Method Scope and Variable Accessibility

Description:
Write a Java program that defines a class with a method `sum` that takes two integer parameters and returns their sum. Inside the `main` method, declare an integer variable `result` and call the `sum` method. Also, create another method named `multiply` that takes two integers and multiplies them, returning the result. Try to access the `result` variable defined in `main` from within the `multiply` method and observe what happens.

Concepts Covered:

- Method scope and variable accessibility
- Understanding that variables declared inside a method are not accessible outside that method

Hint:

- Variables declared inside a method (local variables) are only accessible within that method.

Problem 3: Block Scope and Shadowing

Description:
Write a Java program that defines a method `calculate` which contains a loop. Inside the loop, declare a variable named `x` and assign it a value. Also, declare a variable with the same name `x` outside the loop but inside the method. Print the value of `x` inside and outside the loop to demonstrate block scope and variable shadowing.

Concepts Covered:

- Block scope and shadowing
- Understanding how variables declared inside a block (e.g., a loop or an `if` statement) can shadow variables with the same name in an outer block

Hint:

- Variables declared inside a loop or an `if` statement have block scope and can shadow variables with the same name in the outer method scope.