# Using Static Methods, Public Methods, and Organizing Methods Across Classes in Java

## Project Abstract

The purpose of this project is to demonstrate how to define and use static methods, public methods, and how to organize and call methods across different classes in Java. This project focuses:

1. Define static methods that can be called directly using the class name.
2. Define public methods that require an instance of the class to be invoked.
3. Understand the difference between static and public methods, and how to use them effectively in different scenarios.

## Tasks Overview

### Task 1: Static Method

Objective: Define a static method that can be called directly from the class without creating an object.

Detailed Description: In this task, you will create a class MathOperations and define a static method called squareNumber that accepts an integer and returns its square.

- Steps:

1. Create a class called MathOperations.
2. Define a static method squareNumber(int num) that takes one integer and returns its square.
3. In the main method, call the squareNumber method using the class name without creating an object.

**Note:** You need to print the message in the following format:

```
Square of 5: <result>
```

### Task 2: Public Method

Objective: Define a public method that requires an object to be created for invoking it.

Detailed Description: In this task, you will create a public method in the MathHelper class called addNumbers. This method will accept two integers as parameters and return their sum.

- Steps:

4. Create a class called MathHelper.
5. Define a public method addNumbers(int a, int b) that takes two integers and returns their sum.
6. In the main method of the main class, create an object of the MathHelper class and call the addNumbers method with two integers.

**Note:** You need to print the message in the following format:

```
Sum of 10 and 20: <result>
```

### Task 3: Organizing Methods Across Multiple Classes

Objective: Organize and call methods from multiple classes.

Detailed Description: In this task, you will organize the code into two separate classes (MathOperations and MathHelper) and call methods across these classes from the main class.

- Steps:

7. Define the MathOperations class with the static method squareNumber.
8. Define the MathHelper class with the public method addNumbers.
9. In the main(), test the methods by calling the static method squareNumber directly from the class and the public method addNumbers through an object of the MathHelper class.
10. Create a variable squareResult of type int and assign it value returned from squareNumber(5) being directly invoked using class name i.e MathOperations.
11. Print the result as `"Square of 5: " + squareResult`.
12. Create an object of MathHelper class with name helper. Create a variable sumResult of type int. and assign it value returned from addNumbers(10, 20) being invoked using the helper object.
13. Print the result as `"Sum of 10 and 20: " + sumResult`.

## Execution Steps to Follow:

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**

2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top)  Terminal New Terminal.**
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
6. **To run your project use command:**
   **mvn compile exec:java**
   **-Dexec.mainClass="com.yaksha.assignment.StaticAndPublicMethodsAssignment"**
7. **To test your project test cases, use the command**
   **mvn test**

8. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**