

Using Methods, Parameters, and Return Values in Java

Project Abstract

The purpose of this project is to demonstrate how to define and use methods in Java, pass parameters to methods, and return values from methods. This project focuses:

1. Define methods in a Java class.
2. Use methods to perform operations with parameters.
3. Return values from methods.

Tasks Overview

Task 1: Create a Class Method

Objective: Define a simple class with a method that performs a task, such as returning a string message.

Detailed Description:

In this task, you will create a class called `Calculator` and define a method called `welcomeMessage` that returns a welcome message when called.

- Steps:
 1. Define a class called `Calculator`.
 2. Create a method called `welcomeMessage` that returns a string message.
 3. In the main program, create an object of the `Calculator` class and call the `welcomeMessage` method.

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

```
Welcome to the Calculator!
```

Task 2: Method with Parameters

Objective: Create a method that accepts parameters, processes them, and returns a value.

Detailed Description:

In this task, you will modify the `Calculator` class to include a method called `addNumbers`. This method will accept two integer parameters and return their sum.

- Steps:

4. Add a method `addNumbers(int a, int b)` to the Calculator class.
5. This method should return the sum of the two integers.
6. In the main program, call the `addNumbers` method with two integer values.

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

Sum: <result>

Task 3: Method with Return Value

Objective: Create a method that accepts parameters and returns a value.

Detailed Description:

In this task, you will modify the Calculator class to include a method called `multiplyNumbers`. This method will accept two integer parameters and return their product.

- Steps:

7. Add a method `multiplyNumbers(int a, int b)` to the Calculator class.
8. This method should return the product of the two integers.
9. In the main program, call the `multiplyNumbers` method with two integer values.

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

Product: <result>

Task 4: Invoke methods in main()

Objective: Create an object of Calculator class with name calculator.

Detailed Description:

In this task, you will invoke all above created methods in calculator class i.e `welcomeMessage`, `addNumbers` and `multiplyNumbers`..

- Steps:

10. Create an object of type Calculator with name calculator using new keyword..
11. Invoke `calculator.welcomeMessage` and store its result in String type variable with name message and then print it.
12. Invoke `calculator.addNumbers(10, 20)` and store its result in an int type variable with name sum and then print it as `"Sum: " + sum``.
13. Invoke `calculator.multiplyNumbers(5, 6)` and store its result in int type variable with name product and then print it as `"Product: " + product``.

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

```
"Welcome to the Calculator!"
```

```
"Sum: " + 30
```

```
"Product: " + 30
```

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. To run your project use command:

```
sudo JAVA_HOME=$JAVA_HOME /usr/share/maven/bin/mvn compile exec:java  
-Dexec.mainClass="com.yaksha.assignment.MethodOperationsAssignment"
```

***If it asks for the password, provide password : pass@word1**

5. To test your project test cases, use the command

```
sudo JAVA_HOME=$JAVA_HOME /usr/share/maven/bin/mvn test
```

***If it asks for the password, provide password : pass@word1**