

Runnable Interface Implementation - Assignment

Objective

In this assignment, you will learn how to implement multithreading in Java using the Runnable interface. You will complete a Java program by implementing a custom Runnable class and creating multiple threads to execute tasks concurrently.

Project Setup

Your project structure includes the following:

- A Java class named RunnableExample which contains a main method.
- The main method will contain logic to create and start multiple threads.
- A custom class named MyRunnable that implements the Runnable interface. This class should override the run() method.

Tasks to Complete

You will be provided with a skeleton Java class (RunnableExample.java) where the implementation of the main method and the MyRunnable class should be done by you. Your task is to complete the following:

1. Create and implement the MyRunnable class:
 - The class should implement the Runnable interface.
 - Override the run() method to print a message indicating the thread name that is running as "<thread name> is running".
2. Inside the main method:
 - Create two instances of MyRunnable with name as "runnable1" and "runnable2".
 - Use these instances to create and start two threads.
 - Ensure each thread runs independently and outputs its own message to the console.

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. 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.
5. To run your project use command:

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

***If it asks for the password, provide password : pass@word1**
6. 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**