Task 1: Eclipse IDE Basic Operations

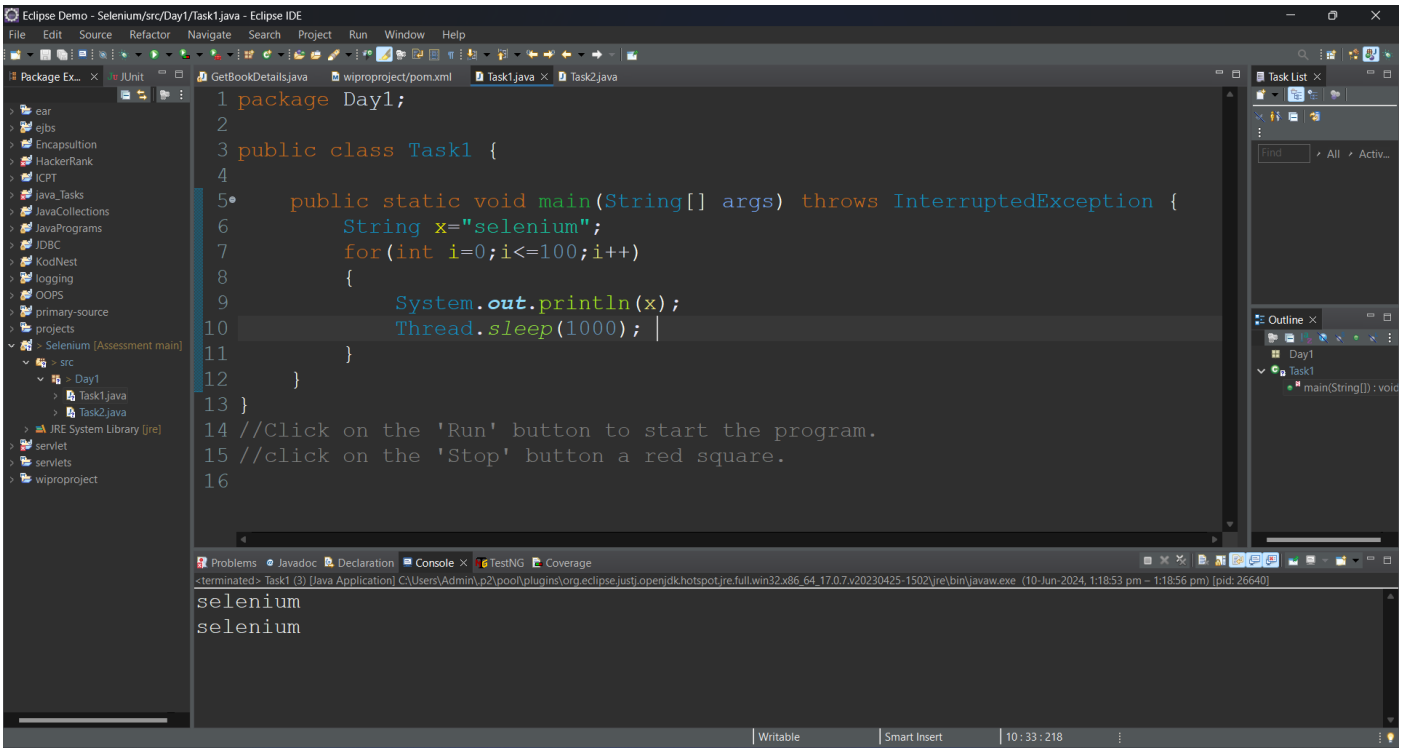
Topic: Navigating the Eclipse Interface

Description: Demonstrate the basic operations in Eclipse by running a pre-written Java program, observing the console output, and then stopping the execution.

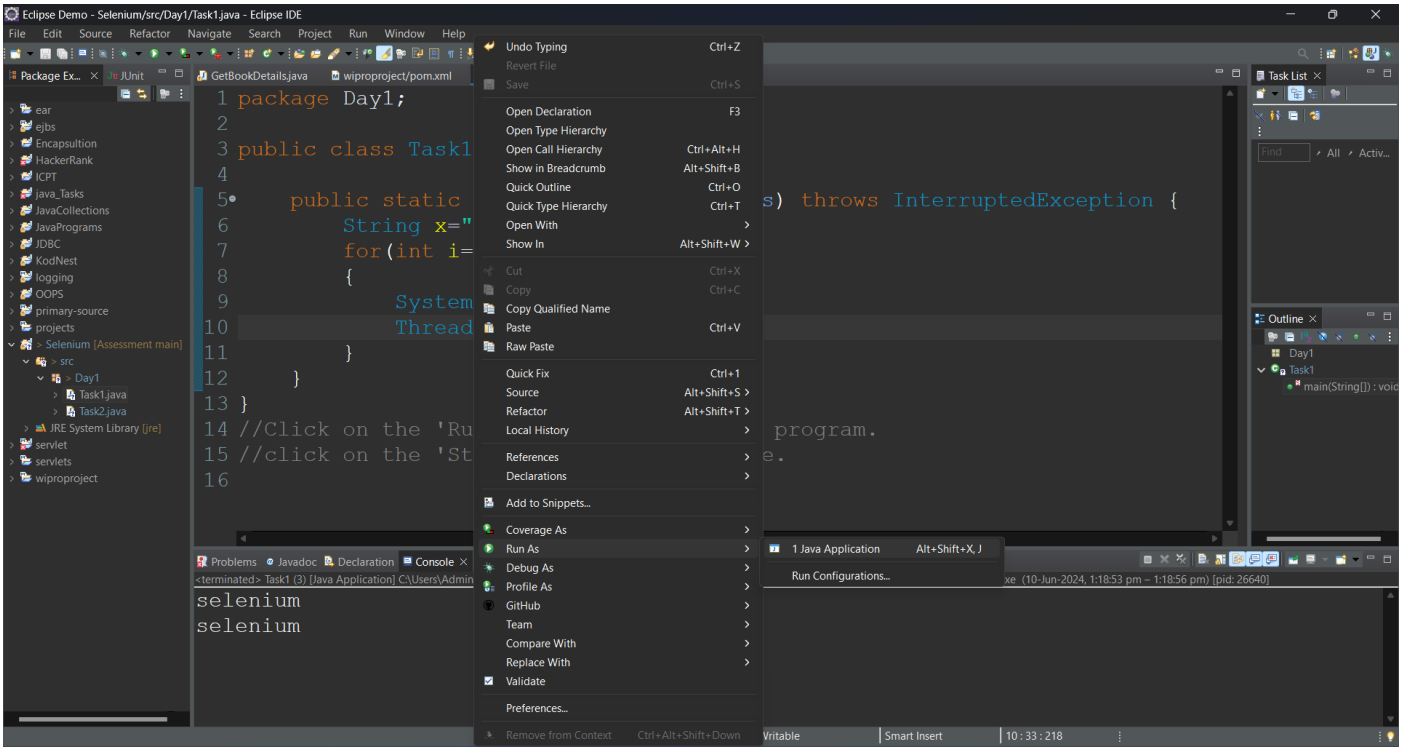
Steps:

Open Eclipse IDE and select the workspace that contains the pre-written Java program.

Locate the Java program in the Package Explorer and open it in the editor.

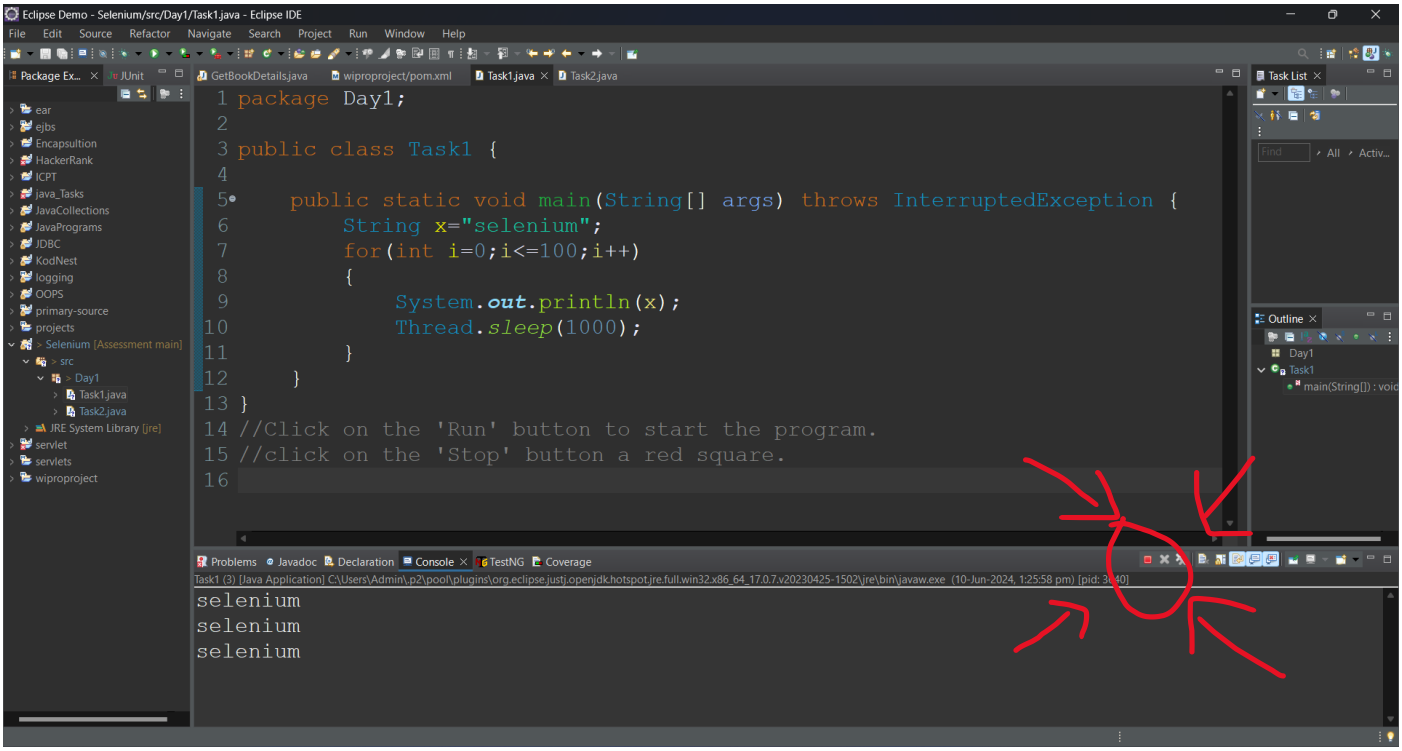


Click on the 'Run' button to start the program and observe the output in the Console window.



Use the 'Stop' button to terminate the program execution.

- click on the 'Stop' red square button.



Describe what the 'Pause' button does in the context of running programs.

Using these buttons helps to control the execution of your Java programs in Eclipse making it easier to develop, test and debug the code.

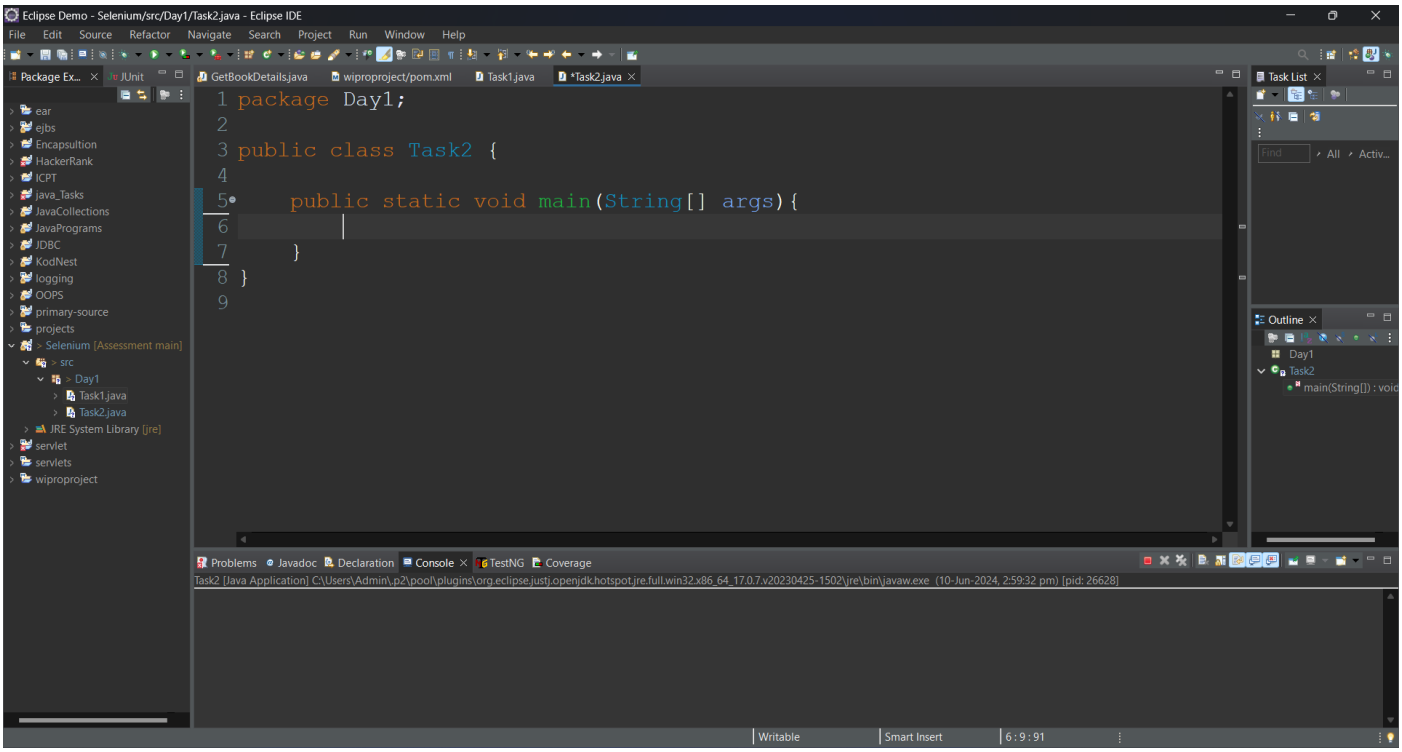
Task 2: Debugging with Eclipse

Topic: Error Handling and Debugging

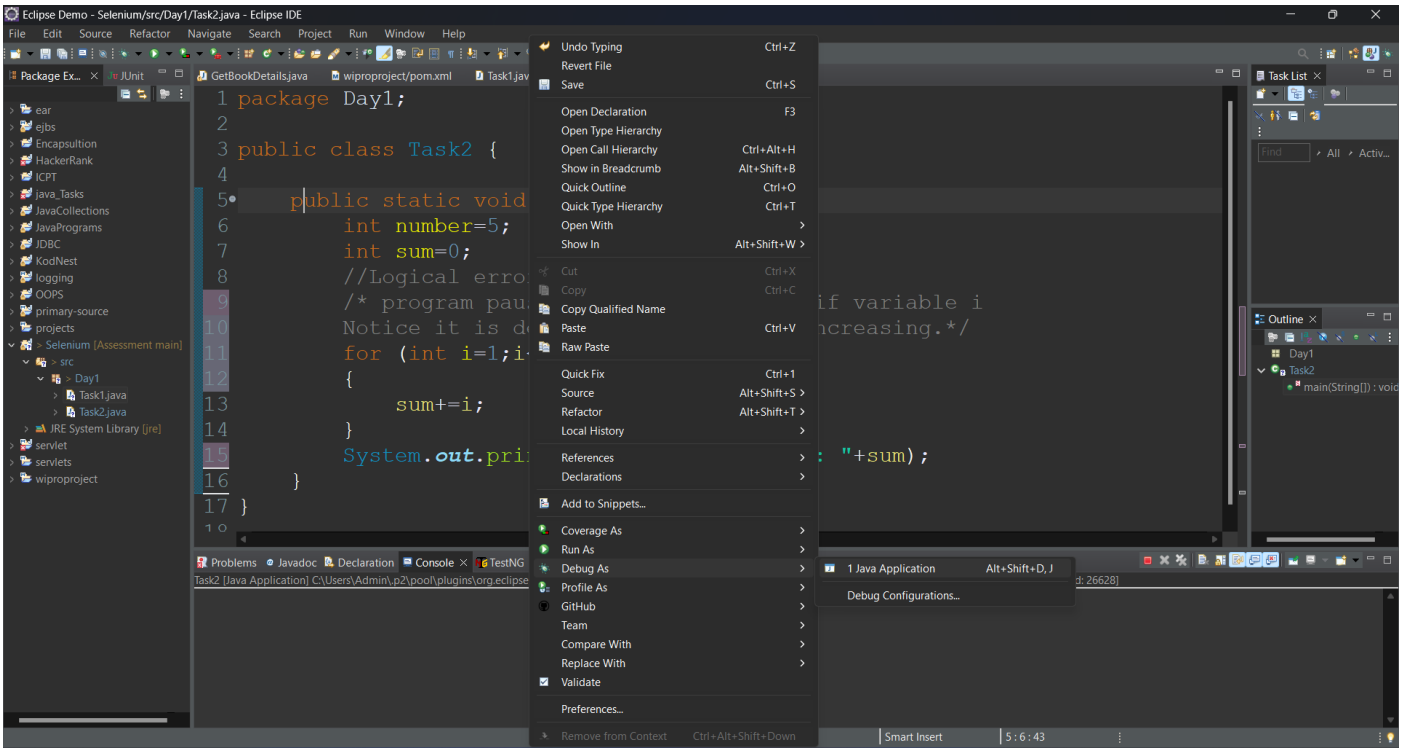
Description: Utilize Eclipse's debugging features to identify and fix a simple logical error in a Java program.

Steps:

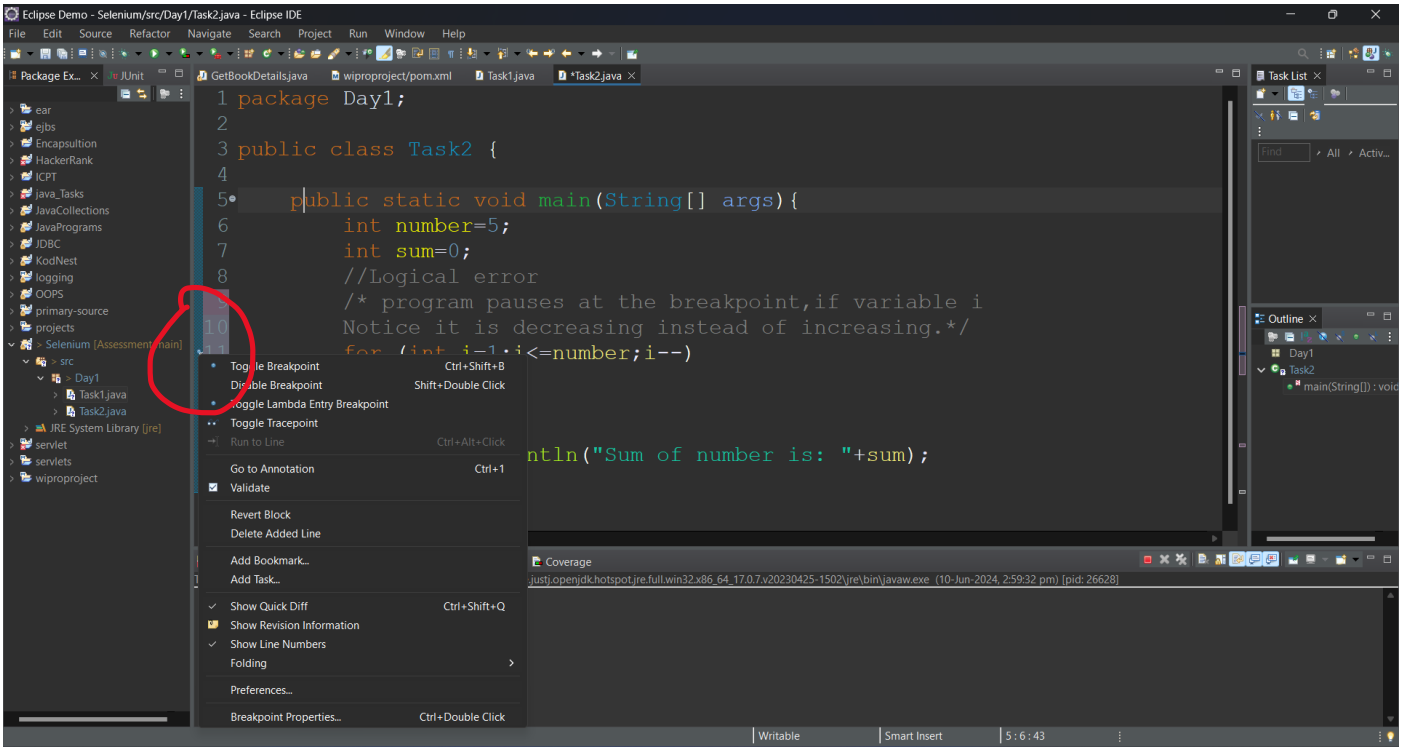
Open the provided Java program with a known logical error in Eclipse.



Start the debugger by clicking on the 'Debug' button.



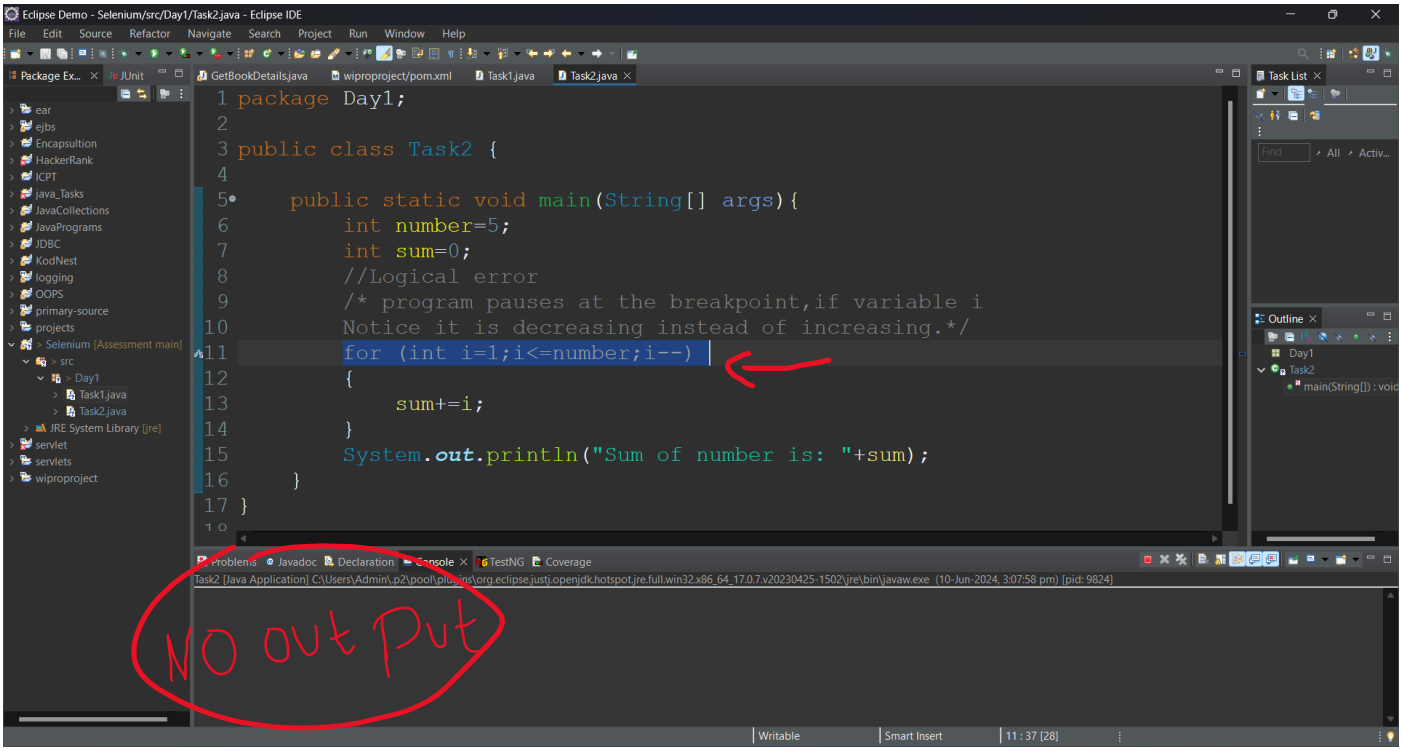
Use breakpoints to pause the execution at critical points in the program.



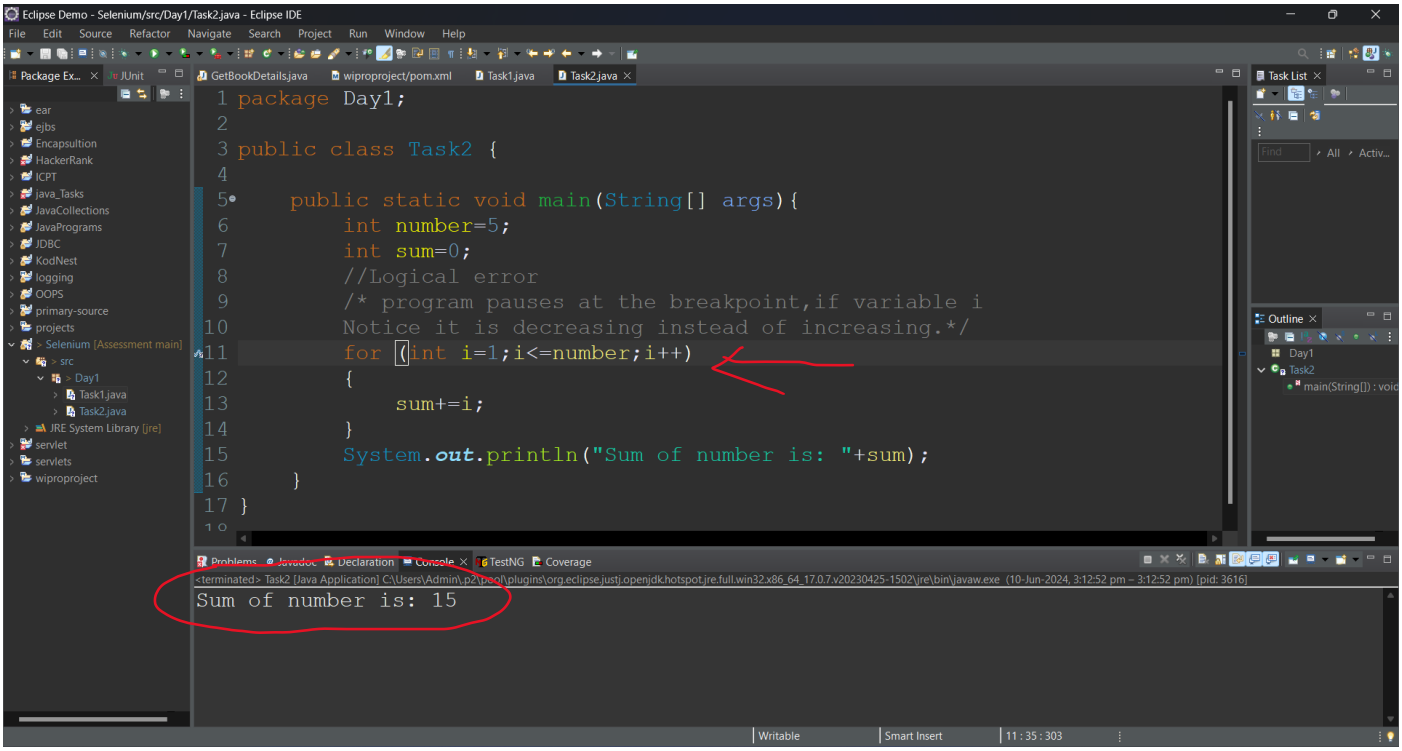
Step through the code using 'Step into' and 'Step over' to observe variable values.

Program pauses at the breakpoint, if variable I notice it is decreasing instead of increasing.

Identify the logical error based on variable values and code flow.



Correct the error and rerun the program to ensure it's fixed.




```java
package Day1;

public class Task2 {

    public static void main(String[] args){
        int number=5;
        int sum=0;
        //Logical error
        /* program pauses at the breakpoint,if variable i
        Notice it is decreasing instead of increasing.*/
        for (int i=1;i<=number;i++)
        {
            sum+=i;
        }
        System.out.println("Sum of number is: "+sum);
    }
}
```

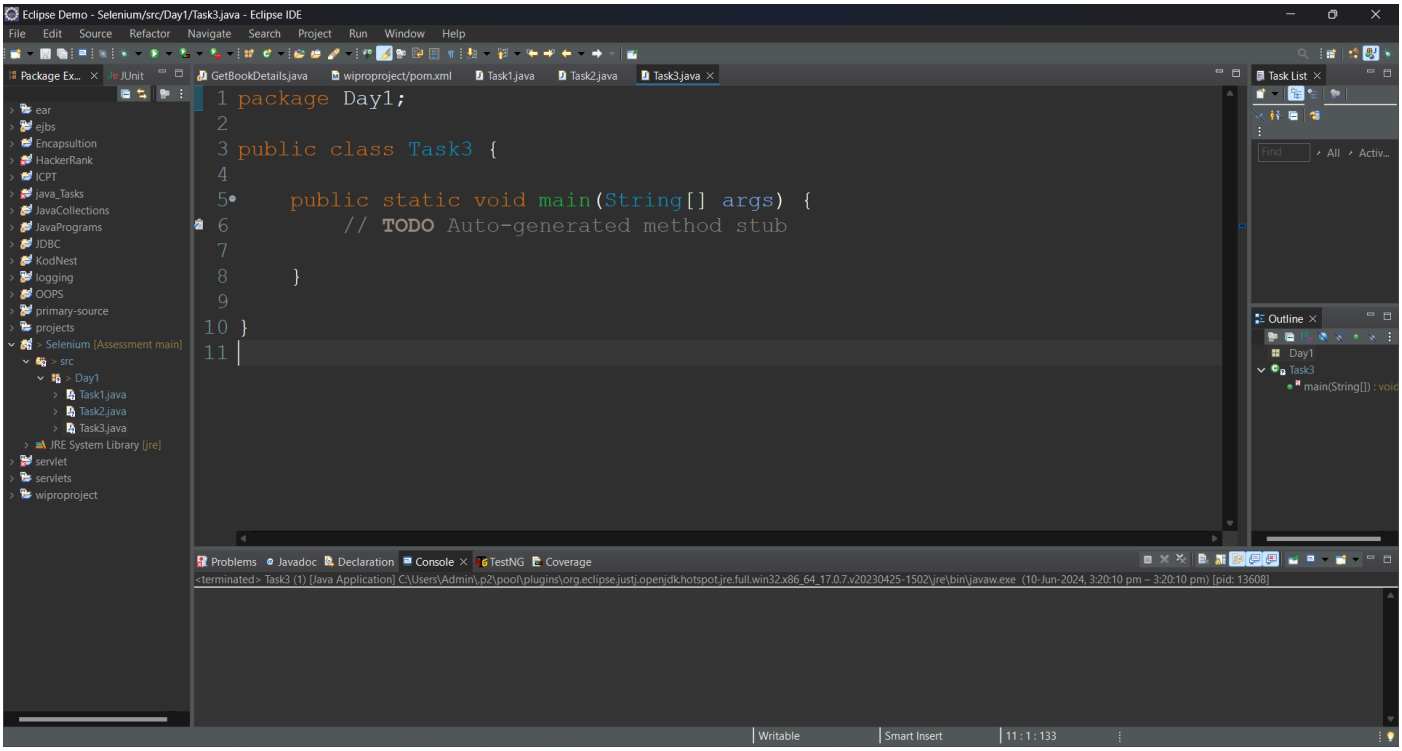
Task 3: Writing and Running a Simple Selenium Test

Topic: First Test Case and WebDriver Basics

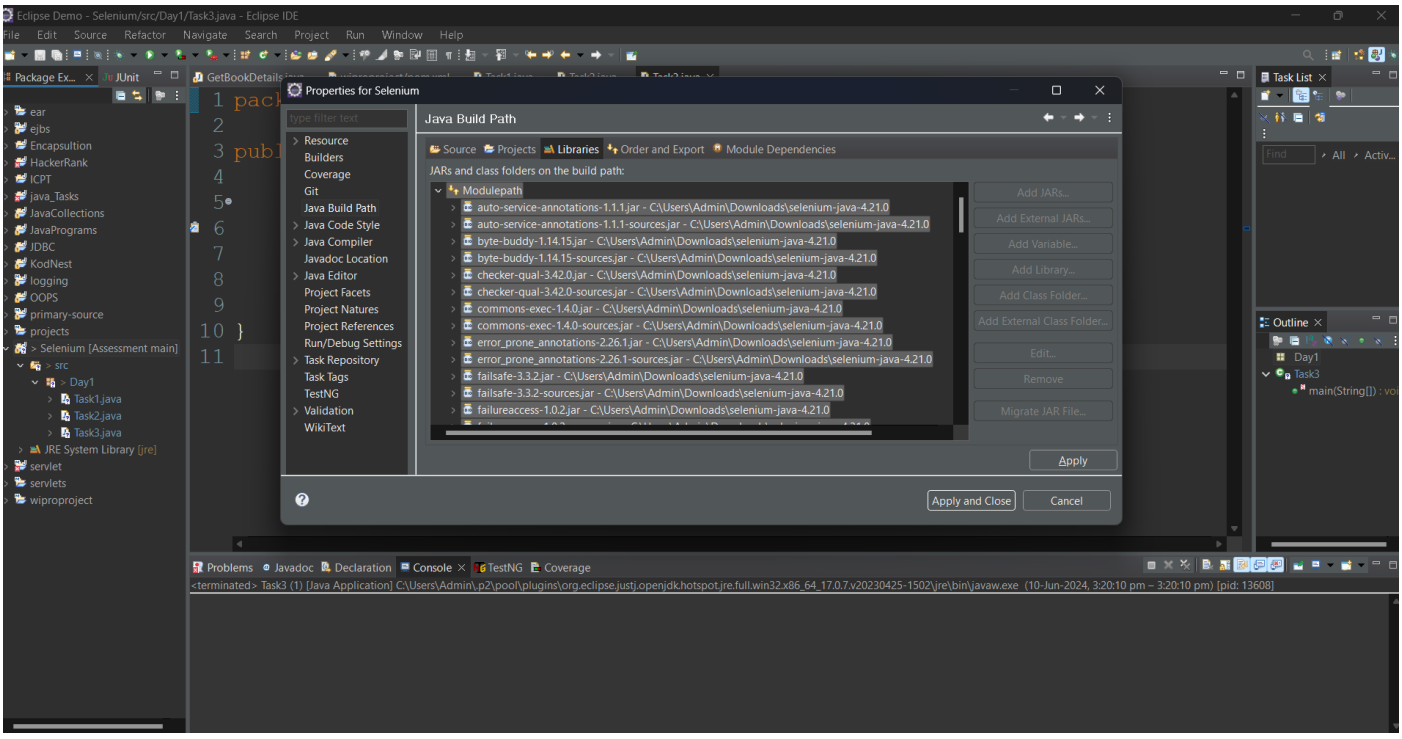
Description: Write a simple Selenium WebDriver test in Eclipse to open a web browser and navigate to a specified URL.

Steps:

Open Eclipse and create a new Java Project dedicated to Selenium tests.



Within the project, set up the Selenium WebDriver by adding the WebDriver JAR files to the build path.



Create a new Java class file for the test case.

Write a Java method using WebDriver to initiate a Firefox, Chrome, or Safari browser session.

Use WebDriver to navigate to 'http://example.com' or a similar simple web page.

```
package Day1;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Task3 {

	public static void main(String[] args) throws InterruptedException

{

		System.setProperty("WebDriver.chrome.driver","C:\\Users\\Admin\\Downloads\\chromedriver-win64\\chromedriver-win64");

	WebDriver driver=new ChromeDriver();

	driver.get("https://www.wipro.com/");

	String title=driver.getTitle();

	System.out.println(title);

	Thread.sleep(4000);

	driver.quit();

	}

}
```

Run the test to ensure the browser opens and navigates to the URL successfully.