

B) In your own language, explain what assertion is and how the “assertThat” method work

Assertion is a statement that verifies that a certain condition holds true. If the condition is false, the test is false. On the other hand, assertThat works with the Hamcrest library, which provides a more readable and expressive way of writing assertions.

The syntax work like this:

`assertThat(actualValue, matcher);`

- **actualValue:** The value you want to test.
- **matcher:** A condition that actualValue is expected to satisfy (e.g., equalTo, greaterThan).

```
package iloveyouboss;

import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import static org.hamcrest.MatcherAssert.assertThat;
import org.junit.*;

public class ScoreCollectionTest {
    @Test
    public void answersArithmeticMeanOfTwoNumbers() {
        // Arrange
        ScoreCollection collection = new ScoreCollection();
        collection.add(() -> 5);
        collection.add(() -> 7);

        // Act
        int actualResult = collection.arithmeticMean();

        // Assert
        assertThat(actualResult, equalTo(6));
    }
}
```

C) In your code, replace assertThat with two alternatives for assertion and run the code again (two separate times).

```
public class ScoreCollectionTest {
    @Test
    public void answersArithmeticMeanOfTwoNumbers() {
        // Arrange
        ScoreCollection collection = new ScoreCollection();
        collection.add(() -> 5);
        collection.add(() -> 7);

        // Act
        int actualResult = collection.arithmeticMean();

        // Assert
        assertEquals(6, actualResult);
    }
}
```

`assertEquals(expected, actual)` compares two values for equality. If they differ, the test fails.

```
public class ScoreCollectionTest {  
    @Test  
    public void answersArithmeticMeanOfTwoNumbers() {  
        // Arrange  
        ScoreCollection collection = new ScoreCollection();  
        collection.add(() -> 5);  
        collection.add(() -> 7);  
  
        // Act  
        int actualResult = collection.arithmeticMean();  
  
        // Assert  
        assertTrue(actualResult == 6);  
    }  
}
```

assertTrue(condition) evaluates a boolean condition. If the condition is false, the test fails.