# Quiz 03 - Unit Testing

LATEST SUBMISSION GRADE

100%

1. If you want to run a setUp method before every test method, what should you write?

1 / 1 point

○ @Test

● @BeforeEach

○ @return

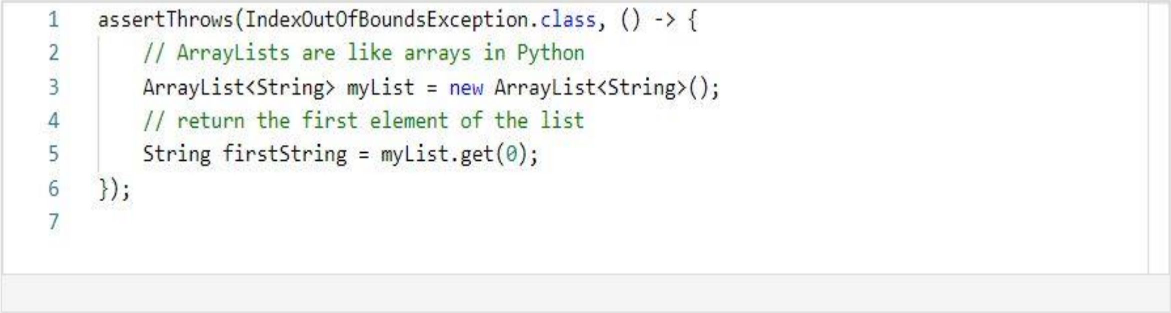
○ @BeforeThis

✓ **Correct**

You should write @BeforeEach as @BeforeEach sets up test data before every test method, similar to python's setUp method.

2. Will this test pass, fail or throw an error?

1 / 1 point



```java
assertThrows(IndexOutOfBoundsException.class, () -> {
    // ArrayLists are like arrays in Python
    ArrayList<String> myList = new ArrayList<String>();
    // return the first element of the list
    String firstString = myList.get(0);
});
```

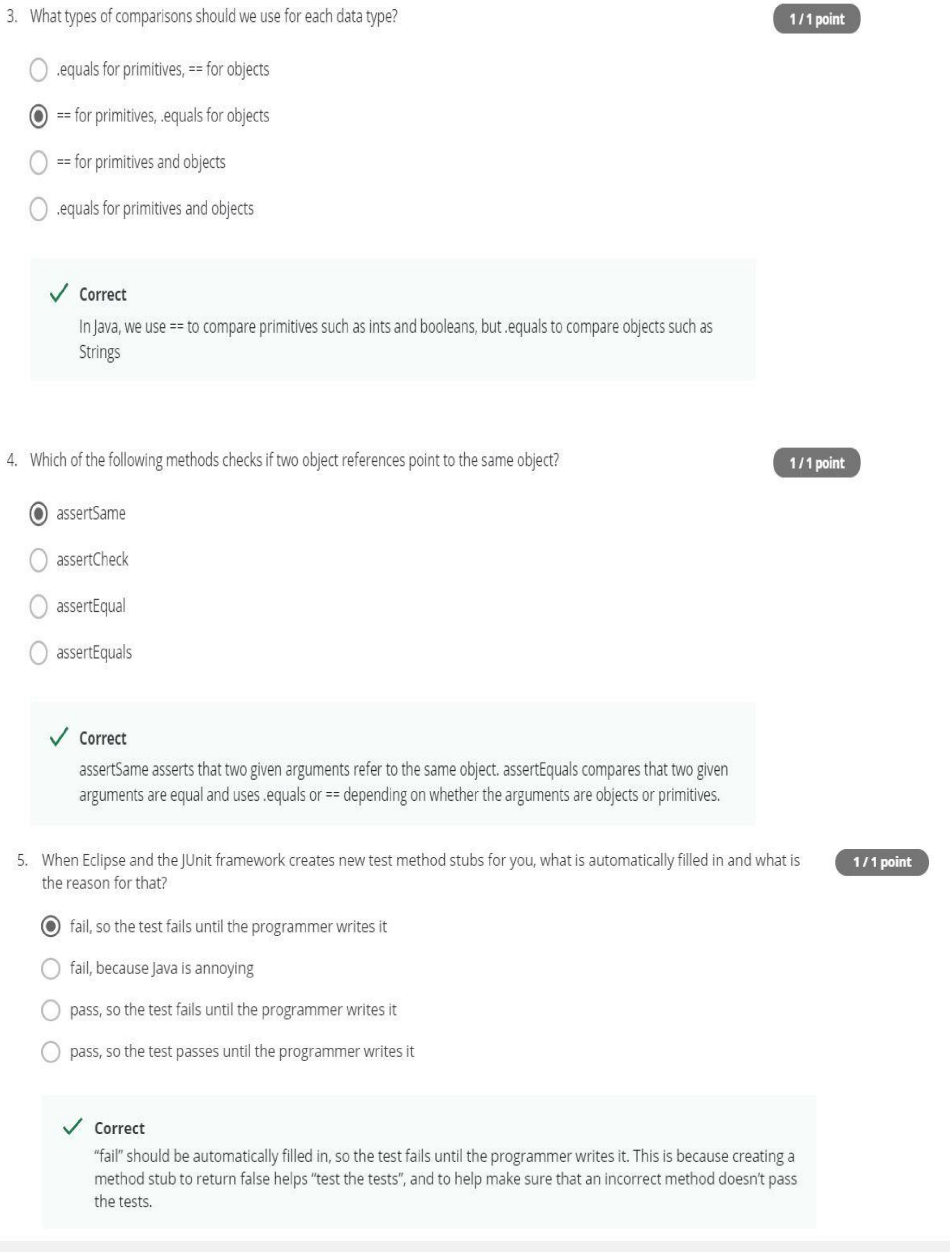
○ throw an error

○ fail

● pass

✓ **Correct**

In assertThrows, we can specify which exception we are expecting. In this case, we are initializing an empty ArrayList and trying to get the first element from the list, which will result in an IndexOutOfBoundsException. Since this is what we were expecting, the test will pass.



3. What types of comparisons should we use for each data type?

1 / 1 point

○ .equals for primitives, == for objects

● == for primitives, .equals for objects

○ == for primitives and objects

○ .equals for primitives and objects

✓ **Correct**

In Java, we use == to compare primitives such as ints and booleans, but .equals to compare objects such as Strings

4. Which of the following methods checks if two object references point to the same object?

1 / 1 point

● assertSame

○ assertCheck

○ assertEqual

○ assertEquals

✓ **Correct**

assertSame asserts that two given arguments refer to the same object. assertEquals compares that two given arguments are equal and uses .equals or == depending on whether the arguments are objects or primitives.

5. When Eclipse and the JUnit framework creates new test method stubs for you, what is automatically filled in and what is the reason for that?

1 / 1 point

● fail, so the test fails until the programmer writes it

○ fail, because Java is annoying

○ pass, so the test fails until the programmer writes it

○ pass, so the test passes until the programmer writes it

✓ **Correct**

“fail” should be automatically filled in, so the test fails until the programmer writes it. This is because creating a method stub to return false helps “test the tests”, and to help make sure that an incorrect method doesn’t pass the tests.