

2. Which of the following statements are true of unit tests? (Select all that apply)

Show Results

20/22 Students Answered

- ☐ A They're typically run by the product owner X
- ☒ B They isolate each part of the program to show that individual parts are correct ✓
- ☐ C They mimic human interaction with the program's user interface X
- ☒ D They're always automated tests ✓
- ☐ E They're primarily used at the end of development, immediately prior to release X
- ☐ i No Explanation

4. A unit test should perform the following steps in what order:

Show Results

20/22 Students Answered

- ☐ A Act, Arrange, Assert
- ☐ B Assert, Act, Arrange
- ☒ C Arrange, Act, Assert
- ☐ D Act, Assert, Arrange
- ☐ i No Explanation

Arrange -- means set up create any object we need
-- set up what you expect to come back

Act -- call the method you are testing and get
back the actual result

Assert -- you assert
Assert.assertEquals

< Back to Results Table

< 5 of 9 >

5. You're testing a Calculator class that has an add method that takes two ints and returns the result of adding them together. If the add method works correctly, what happens when you run the following test?

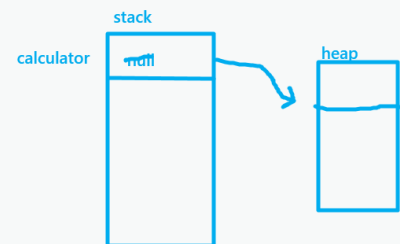
```
@Test
public void add_returns_the_correct_sum() {
    Calculator calculator = null;
    int result = calculator.add(2, 3);
    Assert.assertEquals(5, result);
}
```

= new Calculator();

Show Results

20/22 Students Answered

- ☐ A The test passes
- ☐ B The test fails
- ☒ C The test encounters an error ✓ null pointer exception
- ☐ D The test won't run because it is not a valid JUnit test
- ☐ i No Explanation



6. Now you're testing a subtract method that takes two values and returns the result of subtracting the first value from the second value. Assuming the subtract method works correctly, what happens when you run the following test?

```
@Test
public void subtract_returns_the_correct_difference {
    Calculator calculator = new Calculator();

    int result = calculator.subtract(9, 10);

    Assert.assertEquals(1, result);
}
```

[Show Results](#)

20/22 Students Answered

- ☒ A The test passes ✓
- ☐ B The test fails
- ☐ C The test encounters an error
- ☐ D The test won't run because it is not a valid JUnit test
- ☐ i No Explanation

7. You're writing unit tests for a method called `findMatchingNames` that takes a single char parameter and returns a list of names that begin with that character. The parameter is normally expected to be in the range A-Z.

Which of the following parameter values should you be sure to include tests for? (Select all that apply)

[Show Results](#)

20/22 Students Answered

- ☒ A A and Z ✓
- ☐ B All the characters between A and Z ✗
- ☒ C A character outside the A to Z range ✓
- ☐ D null ✗

8. When the tests in this class are run, how many failing tests will the JUnit report?

[Hide Answers](#)[Show Names](#)

20/22 Students Answered

1

3

1

0

1

1

```
public class DemoTest {

    @Test
    public void this_test_passes() {
        Assert.assertTrue(true);
    }

    public void this_test_fails() {
        Assert.assertTrue(false);
    }

    public void this_test_also_passes() {
        Assert.assertTrue(true);
    }

}
```

0 failing tests


9. You're writing unit tests and notice that the "arrange" step is the same for all of them. So, you factor out the common code and put it into a separate method within the test class. What annotation should that new method have? (include the @)

Hide Answers

Show Names

20/22 Students Answered

@Before annotation

-	
@ Test public void	
I dont know	
@Override public void Assert()	