JUnit Practice KBA

Due Oct 23 at 8am **Points** 21 **Questions** 21
Available Oct 21 at 3pm - Oct 23 at 8am 1 day **Time Limit** None

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	19 minutes	14 out of 21

Correct answers are hidden.

Submitted Oct 23 at 6:54am

Partial

Question 1

0.67 / 1 pts

What are some of the available techniques to test expected exceptions? Choose all that apply.

- [x] Use of 'expected' attribute inside @Test annotation.
- [ ] Try-catch with fail() statement
- [x] Verification with the ExpectedException rule.
- [ ] None of the listed options

Incorrect

Question 2

0 / 1 pts

Which of the following is true about Parameterized test?

- ○ Runs several sets of test data against the same test case.

○ Runs a test once with fixed sets of parameters.

○ It is used to bundle a couple of unit test cases and run them together.

[✓ With Parameterized tests you can test wether the code throws a desired exception or not.]

Incorrect

Question 3

0 / 1 pts

"@AfterClass" annotation can be used only with static methods. State true or false.

[✓ True]

○ False

Question 4

1 / 1 pts

Which statement is true about this program?

Given the class MyClass as follows:

public class MyClass {

public int multiply(int x, int y) {

// the following is just an example

if (x > 999) {

throw new IllegalArgumentException("X should be less than 1000");

}

return x / y; }

}

What would be the outcome of the following test?

```
public class MyClassTest {

@Test(expected = IllegalArgumentException.class)

public void testExceptionIsThrown() {

MyClass tester = new MyClass();

tester.multiply(1000, 5); }

}
```

- ○ The testExceptionIsThrown() test cases executes without any exception.
- ○ The testExceptionIsThrown() test case fails.
- [✓ The testExceptionIsThrown() test case is successful.]
- ○ The program has syntax errors.

Incorrect

Question 5

0 / 1 pts

Which of the following Hamcrest matchers tests for null value?

- [✓ isNull]
- ○ nullValue
- ○ isNullValue
- ○ isNullable

Partial

Question 6

0.67 / 1 pts

Which of the following are some good reasons to use Mock objects in Junit testing? Choose all that apply.

- [✓] The real obbject has non-deterministic behaviour and is difficult to set up.
- [ ] None of the listed options.
- [ ] The real object is not yet available.
- [✓] The real object is too complex to be used in a unit test because it depends on external resources.

Incorrect

Question 7

0 / 1 pts

What would be the output if we run the following code?

```
import org.junit.BeforeClass;

import org.junit.Test;

public class Sample1 {

@BeforeClass

public void beforeClass(){

System.out.println("before class"); }

@Test

public void test1(){

System.out.println("test1"); }

}
```

- ○ Code compiles correctly but throws Exception during runtime.
- ● Compilation error
- ○ Code compiles, runs successfully and prints: test1
- ○ Code compiles, runs successfully and prints: before class test1

Question 8

1 / 1 pts

A Junit test method should be public.

- [● True]
- ○ False

Question 9

1 / 1 pts

What information does an object of the Failure class contains?

- ○ Only the description of the failed test.
- [● Description of the failed test and the exception thrown while running it.]
- ○ Description of the fail() method
- ○ Description of the exception thrown.

Partial

Question 10

0.67 / 1 pts

What are some of the best practices for Unit Testing? Choose all that apply.

- [✓ Write a separate test class for each class that needs to be tested.]
- ☐ Declare test methods as private static.
- [✓ Write at least two unit test cases for each requirement: one positive test and one negative test.]

- [✓] Mock out external services or state.
- [✓] Make each test not dependent on other tests.

Question 11

1 / 1 pts

Which of the following options asserts that List items are in given order?

- [✓ ●]
  List items = Arrays.asList("item1", "item2", "item3", "item4");
  assertThat(hamcrestMatchers, contains("item1", "item2", "item3", "item4"));
- ○
  List items = Arrays.asList("item1", "item2", "item3", "item4");
  assertThat(hamcrestMatchers, containsInAnyOrder("item3", "item1", "item2", "item4"));
- ○
  List items = Arrays.asList("item1", "item2", "item3", "item4");
  assertThat(hamcrestMatchers, hasItems("item4", "item3", "item2", "item1"));
- ○
  List items = Arrays.asList("item1", "item2", "item3", "item4");
  assertThat(hamcrestMatchers, hasItems("item1", "item2", "item3"));

Incorrect

Question 12

0 / 1 pts

Which of the following are conventions suggested by the JUnit framework?

- ○ Name of the method must start with "test"
- ○ Return type of the test method must be void.
- ○ Test methods must not have any parameter.

All of the mentioned.

[✓ Name of the class must end with "Test"]

Question 13

1 / 1 pts

What are the values of each of the member variables?

```
@RunWith(Parameterized.class)

public class ParameterizedTestFields {

// fields used together with @Parameter must be public

@Parameter(0) public int m1;

@Parameter(1) public int m2;

@Parameter(2) public int result;

// creates the test data

@Parameters public static Collection data() {

Object[][] data = new Object[][] { { 1 , 2, 2 }, { 5, 3, 15 }, { 121, 4, 484 } };

return Arrays.asList(data);

}

@Test public void testMultiplyException() {

MyClass tester = new MyClass();

assertEquals("Result", result, tester.multiply(m1, m2));

}

// class to be tested

class MyClass {

public int multiply(int i, int j) {

return i *j;}

} }
```

I. m1 = 1, 5, 121 m2 = 2, 3, 4 result = 2, 15, 484
II. m1 = 0, 0, 0 m2 = 0, 0, 0 result = 0, 0, 0
III. m1 = 1, 2, 2 m2 = 5, 3, 15 result = 121, 4, 484

- [✓ I]
- II
- III

Question 14

1 / 1 pts

Which of the following methods of Assert class checks that a condition is true?

- assertBoolean
- [✓ assertTrue]
- assertCheck
- assertChecks

Question 15

1 / 1 pts

Junit TestSuite is a/an:

- [✓ Container class, which is used to group multiple test cases into a collection and run them together.]
- Interface, which contains the declaration of all the methods and needs to be implemented by Test cases.

- ○ Defines test fixture, which contains all the test methods.
- ○ None of the listed options.

Question 16

1 / 1 pts

Which of the following are needed in order to write a Theory in Junit? Choose all that apply.

- ✓ The class should be annotated with @RunWith(Theories.class)
- ✓ The class should have a data method that generates and returns test data, by annotating static member variables with @Datapoint.
- ✓ A test method with @Theory annotation.
- ☐ A public static method that returns a collection of objects with @Parameters annotation.

Question 17

1 / 1 pts

Which of the following statements is true regarding Test Fixture?

- ○ There are two class-level fixture and two method-level ones.
- ○ The purpose of fixture is to provide a fixed environment in which tests are run so that results are repeatabe.
- ○ Test Fixture can help to setup mock objects
- ● All of the listed options

Question 18

1 / 1 pts

Which of the following methods of Assert class checks that two object references are not pointing to the same object?

- ○ void assert(Object expected, Object actual, boolean isSame)
- ○ void assertCheck(Object expected, Object actual, boolean isSame)
- [✓ void assertNotSame(Object expected, Object actual)]
- ○ void assertChecks(Object expected, Object actual, boolean isSame)

Question 19

1 / 1 pts

Choose the appropriate @SuiteClasses annotation to run test classes Test1.class and Test2.class together.

- [✓ "@SuiteClasses(value={Test1.class, Test2.class})"]
- ○ "@SuiteClasses(value=All)"
- ○ "@SuiteClasses(Test1, Test2)"
- ○ "@SuiteClasses()"

Incorrect

Question 20

0 / 1 pts

Which test case(s) will be reported?

public interface FastTests { /* category marker */ }

public interface SlowTests { /* category marker */ }

```
public class A {

@Test

public void a() {

fail();

}

@Category(SlowTests.class)

@Test

public void b() { }

}

@Category({ SlowTests.class, FastTests.class })

public class B {

@Test

public void c() { }

}

@RunWith(Categories.class)

@IncludeCategory(SlowTests.class)

@SuiteClasses({ A.class, B.class }) // Note that Categories is a kind of Suite

public class SlowTestSuite {

}
```

- All test cases.
- No test case will be reported.
- Only the test b() from class A will be reported.
- [✓ All test cases from class B and Only the b() test case from class A will be reported.]

Question 21

1 / 1 pts

How many times would an @Before annotated method get executed if there are three test methods in the test class?

- [✓ 3]
- 1
- 0
- 2