

Define the difference between black box and white box testing:

Black box: Data driven. The focus is on finding cases where the program does not behave according to the design specifications. No knowledge of the internal program behaviour.

White box: Test via examination of the program's logic.

What is the 'gold standard' for black box and white box testing?

Black box – Exhaustive input testing.

White box – Exhaustive Path Testing.

Q1 : Creating a java test class that is called TestJUnit.java. Adding expected exception to testPrintMessage() test case.

```
@Test(expected = ArithmeticException.class)
public void testPrintMessage() {
    System.out.println("Inside testPrintMessage()");
    messageUtil.printMessage();
}
```

Q2-1: The ExpectedException rule allows you to specify, within your test, what exception you are expecting and even what the exception message is. Try to add the expectedexception to test a test case, if you give a negative age in new Person();

```
@Rule
public ExpectedException exception = ExpectedException.none();

@Test
public void testExpectedException() {
    exception.expect(IllegalArgumentException.class);
    exception.expectMessage(("Invalid age"));
    new Person("Joe", -1);
}
```

Q2-2: Specify the exception in the @Test annotation

```
@Test(expected = IllegalArgumentException.class)
public void testExpectedException2() {
    new Person("Joe", -1);
}
```

Q2-3: Use a try-catch block

```
@Test
public void testExpectedException3() {
    try {
        new Person("Joe", -1);
        fail("Should have thrown an IllegalArgumentException because age is invalid!");
    } catch (IllegalArgumentException e) {
        assertEquals(e.getMessage(), ("Invalid age:-1"));
    }
}
```

Q3: The performance verification is one of the most difficult issues in JUnit. JUnit 4 not completely solves the problem but offers an utile help: in fact, the test method can be signed with a timeout parameter. If the tests are running in a time over, it fails. Try to use set the timeout value = 500 and test run() function in Performace.java.

```
@Test(timeout=500)
```