In JUnit, what does a method annotated with the **@AfterAll** annotation signify?

- ☐ a. Method to run before each test case.
- [✓ b. Method to run once after all the tests within a test class.]
- ☐ c. Method to set up resources before testing.
- ☐ d. Method to assert test results.
- ☐ e. Method to run after each test case.

Which of the next test case input sets ensure a complete path coverage for the method bellow?

```java
public static int m(int a, int b) {
  int r;
  if (a > 0) {
    if (b > 0) {
      r = 1;
    } else {
      r = 2;
    }
  } else {
    r = 1;
  }
  return r;
}
```

☐ a.

1. a = -100, b = 102
2. a = 100, b = 105
3. a = 105, b = 107

☐ b.

1. a = -100, b = 102
2. a = 105, b = -107

[redacted] c.

1. a = -55, b = 60
2. a = 70, b = 53
3. a = 65, b = -53

[redacted] d.

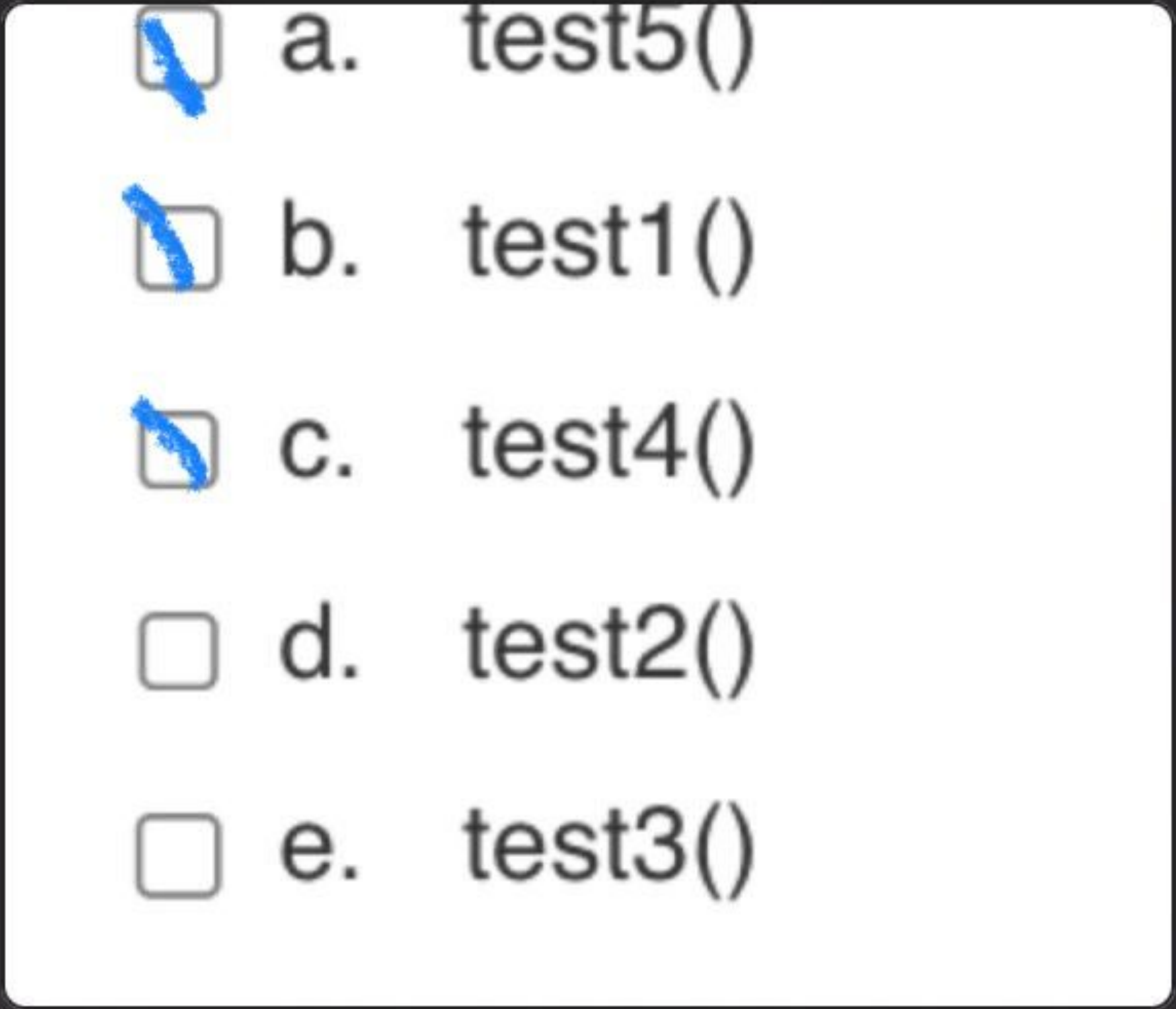
1. a = -100, b = 102
2. a = 100, b = 105
3. a = 105, b = -107

☐ e.

1. a = -100, b = 102
2. a = 100, b = 105

Given the following code, which of the tests will be considered as **passed** by jUnit?

```
class Calculator {
  public double multiply(double a, double b) {
    return a * b;
  }
}
class CalculatorTest {
  private Calculator calculator = new Calculator();
  @Test
  public void test1() {
    assertNotNull(calculator);
  }
  @Test
  public void test2() {
    assertNull(calculator);
  }
  @Test
  public void test3() {
    assertTrue(20.0 == calculator.multiply(15, 5));
  }
  @Test
  public void test4() {
    assertTrue(20.0 == calculator.multiply(20, 1));
  }
  @Test
  public void test5() {
    assertEquals(2, 2 * 1);
  }
}
```



- [✓ a. test5()]
- [✓ b. test1()]
- [✓ c. test4()]
- d. test2()
- e. test3()

Given the test class below, select which of the following statements are true after running the entire test class.

```
class JUnitTest {
 @AfterAll
 public static void clean() {
  System.out.println("After all");
 }
 @AfterEach
 public void tearDown() {
  System.out.println("After each");
 }
 @Test
 public void test1() {
  System.out.println("Test 1 starting");
  assertTrue(true);
  System.out.println("Test 1 ending");
 }
 @Test
 public void test2() {
  System.out.println("Test 2 starting");
  assertFalse(true);
  System.out.println("Test 2 ending");
 }
 @Test
 public void test3() {
  System.out.println("Test 3 starting");
  assertEquals(2, 2*1);
  System.out.println("Test 3 ending");
 }
}
```

- ☐ a. The "After each" messages will be printed 1 time.
- ☐ b. All messages from the 3 test methods (test1(), test2(), test3()) will be printed.
- [✓ c.] The "After all" messages will be printed 1 time.
- [✓ d.] The "After each" message will be printed 3 times.
- ☐ e. The "After all" message will be printed 3 times.

Which of the following are characteristics of black-box testing?

- ☐ a. It identifies coding syntax errors.
- [✓ b. It tests based on inputs and expected outputs derived from the tested entity specification.]
- ☐ c. It requires knowledge about the source code of the tested entity.
- ☐ d. It tests an entity based on its implementation details.
- ☐ e. It tests based on inputs and expected outputs derived from the source code of the tested entity