JUnit Testing – Lab Worksheet

**Name:** Jackie Lao **Period:** 6

**Test for mystery1:**

**What types are the method parameters for mystery1? What is the method return type?**

The method parameters’ types are integer. The method return type is integer.

**Read the comment. What is the mystery1 method supposed to do?**

Mystery1 method is supposed to compute the greatest common factor.

**Are there any initial conditions required for the test? How would you set up these initial conditions?**

Yes, you need to create the object Mystery. To set up the object mystery you can use

Mystery m = new Mystery(“Test1”)

**What actions do you need to carry out in your test? What method calls do you need to make?**

You need to make assertions that compare the expected value and the actual value. You need to call the mystery1 method in the Mystery class.

**What are the expected results of carrying out your actions? What assertions can you make? Make sure to include at least 10 assertions.**

assertEquals(2,m.mystery1(10,4));

assertEquals(4,m.mystery1(28,16));

assertEquals(8,m.mystery1(40,32));

assertEquals(10,m.mystery1(50,90));

assertEquals(144,m.mystery1(144,288));

assertEquals(100,m.mystery1(10000,100));

assertEquals(9,m.mystery1(1800,9));

assertEquals(950,m.mystery1(2850,47500));

assertEquals(105000,m.mystery1(525000,210000));

assertEquals(2,m.mystery1(2,2));

**Run the test. What are the test results? Which assertions failed?**

All the test results passed.

**What conclusions can you make about the method? What might cause a bug in this method? Is it entirely bug-free? If your test failed, why did it fail? What evidence supports your conclusion?**

The method checks to see if the expected value is the same as the actual value and if it is not the same the test will fail. If int a or int b is negative or if the expected value is not the same as the actual value. It is not entirely bug-free. The actual value and the expected value is the same so the test results passed.

**Test for mystery2:**

**What types are the method parameters for mystery2? What is the method return type?**

The method parameters’ types are double. The method return type is double.

**Read the comment. What is the mystery2 method supposed to do?**

The method mystery2 is supposed to solve for the larger x value.

**Are there any initial conditions required for the test? How would you set up these initial conditions?**

Yes, you need to create the object Mystery. To set up the object mystery you can use

Mystery m = new Mystery(“Test2”)

**What actions do you need to carry out in your test? What method calls do you need to make?**

You need to make assertions that compare the expected value and the actual value. You need to call the mystery2 method in the Mystery class.

**What are the expected results of carrying out your actions? What assertions can you make? Make sure to include at least 10 assertions.**

assertEquals(1,m.mystery2(1,2,-3),0.01);

assertEquals(.75,m.mystery2(4,5,-6),0.01);

assertEquals(.71,m.mystery2(6,7,-8),0.01);

assertEquals(.68,m.mystery2(8,9,-10),0.01);

assertEquals(.67,m.mystery2(10,11,-12),0.01);

assertEquals(.62,m.mystery2(89,90,-91),0.01);

assertEquals(.62,m.mystery2(91,92,-93),0.01);

assertEquals(.62,m.mystery2(93,94,-95),0.01);

assertEquals(.63,m.mystery2(95,96,-97),0.01);

assertEquals(.62,m.mystery2(97,98,-99),0.01);

**Run the test. What are the test results? Which assertions failed?**

All the tests results passed.

**What conclusions can you make about the method? What might cause a bug in this method? Is it entirely bug-free? If your test failed, why did it fail? What evidence supports your conclusion?**

The method checks to see if the expected value is the same as the actual value and if it is not the same the test will fail. If double c is not negative or if the expected value is not the same as the actual value. It is not entirely bug-free. The actual value and the expected value is the same so the test results passed.

**Remember to include your JUnit test files!** (Put them in this same folder and make sure they are included in your pull request.)