JUnit Testing – Lab Worksheet

**Name: Jessie Lin Period: 6th**

**Test for mystery1:**

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

The method parameters for mystery 1 are int. The return type is int.

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

The mystery1 supposed to compute the greatest common factor of two integers.

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

A mystery object needs to be created. We can set up the Mystery object using

Mystery t = new Mystery (“object”);

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

We need to call the mystery1 method in the Mystery class. State an assertion that will compare the expected value with the actual value from the mystery1 method.

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

*assertEquals*(10,t.mystery1(10,70) );

*assertEquals*(2,t.mystery1(8,6) );

*assertEquals*(3,t.mystery1(6,9) );

*assertEquals*(7,t.mystery1(42,49) );

*assertEquals*(7,t.mystery1(21,7) );

*assertEquals*(4,t.mystery1(20,12) );

*assertEquals*(6,t.mystery1(24,18) );

*assertEquals*(4,t.mystery1(8,4) );

*assertEquals*(5,t.mystery1(10,15) );

*assertEquals*(16,t.mystery1(32,16) );

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

The test results are all correct. There is no assertions fails.

**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?**

All you have to do is to creat a mystery object and to find the GCF of the numbers. If you do not write down anything for the mystery argument, it will cause a bug. It is entirely bug-free. If the test failed, maybe you write down the common factor and not the common greatest factor. Like if you have "8, 12", the GCF should be 4 and not 2.

**Test for mystery2:**

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

The method parameters for mystery2 are double. The method return type is double.

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

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

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

A mystery object needs to be created. We can set up the Mystery object using

Mystery t = **new** Mystery("object2");

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

We need to call the mystery2 method in the Mystery class. State an assertion that will compare the expected value with the actual value from the mystery2 method.

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

*assertEquals*(-1.0, t.mystery2(1.0, 2.0, 1.0), 0.0);

*assertEquals*(-0.2,t.mystery2(5.0,6.0,1.0),-1);

*assertEquals*(3.0,t.mystery2(1.0,4.0,-21.0),-7);

*assertEquals*(3.0,t.mystery2(1.0,-5.0,6.0),2.0);

*assertEquals*(3.0,t.mystery2(1.0,-5.0,6.0),4.0);

*assertEquals*(-0.5,t.mystery2(2.0,7.0,3.0),-3.0);

*assertEquals*(2.0,t.mystery2(1.0,2.0,-8.0),-4.0);

*assertEquals*(2.0,t.mystery2(1.0,-4.0,4.0),2.0);

*assertEquals*(1.0,t.mystery2(1.0,4.0,-5.0),-5.0);

*assertEquals*(1.0,t.mystery2(1.0,2.0,-3.0),-3.0);

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

The test results are all correct. There is no assertions failed.

**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?**

You need to creat a mystery object and it returns the larger number. If t.mystery() do not have three arguement in it, it might cause a bug in the method. It is entirely bug-free. If the test failed, you need to check the math you did and the numbers order. If you have the number 1,4, and -5, the first number(root1) you need to write down is 1 instead of -5; because root1 gives you 1 and root2 gives you -5.

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