JUnit Testing – Lab Worksheet

**Name: Shirley Mei Period: 6°**

**Test for mystery1:**

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

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

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

The myster1 method is supposed to compute the greatest common divisor of two integers, ‘a’ and ‘b’ and return the GCD.

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

The initial conditions required are to create a Mystery object. Set up these conditions Mystery one = new Mystery (“an object”).

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

Call the mystery1 method and use an assertEquals statement to compare the expected value to the actual value from the mytery1 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, one.mystery1(2, 3));

*assertEquals*(5, one.mystery1(10, 5));

*assertEquals*(9, one.mystery1(9,9));

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

*assertEquals*(12, one.mystery1(60, 72));

*assertEquals*(1, one.mystery1(11, 1));

*assertEquals*(2, one.mystery1(2, 6));

*assertEquals*(11, one.mystery1(66, 55));

*assertEquals*(1, one.mystery1(53, 30));

*assertEquals*(49, one.mystery1(-49,49));

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

The last assertion 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?**

The method can’t include negative numbers. If a number is negative, it will loop forever.

**Test for mystery2:**

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

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

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

Mystery2 is supposed to find and return the larger root of the equation ax2 +bx+c=0.

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

The initial conditions required are to create a Mystery object. Set up these conditions Mystery two = new Mystery (“another object”).

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

Call the method2 object and use the assertEquals statement to test the code.

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

*assertEquals*(2.41, two.mystery2(1, -2, -1), 0.01);

*assertEquals*(0.67, two.mystery2(5, 4,-5), 0.01);

*assertEquals*(4.79, two.mystery2(-1, 5 ,-1), 0.01);

*assertEquals*(0.61, two.mystery2(1, 1 ,-1), 0.01);

*assertEquals*(1.32, two.mystery2(7 ,-10, 1), 0.01);

*assertEquals*(2.11, two.mystery2(-3, 4 , 5), 0.01);

*assertEquals*(1.0, two.mystery2(2, 3 ,1), 0.1);

*assertEquals*(0.42, two.mystery2(4, 3, -2), 0.01);

*assertEquals*(0.19, two.mystery2(-3, -1 , -2), 0.01);

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

The last assertion fails; it returns an error.

**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?**

An assertion will fail if the x-value doesn’t cross the x-axis or if the equation returns an imaginary number.

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