Logan Maier Activity 1

1. Common coding courtesy to write the whole name of the variable so anyone coming to look at the code for the first time will be able to read the code more clearly.
2. Adding “<< endl” ends the line of text after the area is printed in the command line.
3. They are not noticed until the code tries to call something that is abbreviated by the namespace for the convenience of the coder.
4. Method 1: Set radius equal to 1, area expected is pi. Method 2: Set radius equal to 10, area expected is pi*100. In the To-Do list, everything sounds familiar, but I believe I could do 1,2, and 7 with very little assistance.

6. No semicolons, no declarations needed before variable names to set the type, and you can use radius**2 to square instead of radius*radius.

Overflows, Underflows, and Machine Precision:

1. The output is directed to the “ofstream” which in this case prints the statements to the screen and prints them to a file.

1.1 For single-precision, Underflow occurred at power of -45, and overflow occurred at power of 38.

1.2 For double-precision, Underflow occurred at power of -324, and overflow occurred at power of 307.

2.1 Single-Precision floating point numbers, precision = 5.705318884e-08

2.2 Double-Precision floating point numbers, precision = 1.05298825941394181864537e-16

3. The machine precision is how precisely our machines can calculate a number with certain error in the calculation. Bad computers would have large uncertainties in the output, but the more precise, the more reliable the machine.

Using a Makefile

2. Yes it appears to have compiled area.cpp, linking it to area.x.
3. Yes, linked properly.

Using GSL Scientific Library

1. We can see on the first graph that at x = 5, the graph is around -0.2.
2. Our code outputs at x = 5, y = -.177. I believe that this is good indicator our code is working.
3. Answer at x =3 is equal to -0.26
4. Yes, the code successfully output the correct answer.
5. We did not have a chance to talk about this in class.