**Assignment 6 – Templates and Exceptions**

Submit to MUOnline as a compressed (.zip) file containing your code project.

|  |  |  |
| --- | --- | --- |
| 1. | Write a function that takes an integer and returns its integer square root. The function should **throw an exception** if it is passed an integer that is not a perfect square.  For example,  16 is a perfect square since its square root is a whole number, 4.  15 is not since its square root, 3.873… is not a whole number.  Test the function with a perfect square (like 16) and a non-perfect square (like 15). The program must handle the exception properly – it must continue and not crash when you use a non-perfect square. | 40% |
|  |  |  |
| 2. | Use Recursion to write and test a function that raises a number to a power without using any built in pow function, and **without using loops.** (Hint: For the base case, what is any number raised to the 0th power?)  For example,  cout << LooplessPow(2, 4) << endl;//prints out 16 | 30% |
|  |  |  |
| 3. | Write a **function template** that will take 3 objects and return the largest one. Test the function with ints, doubles, and strings in your program. When testing with strings the ‘largest’ should be the last one alphabetically (Zebra is larger than Aardvark). Make sure you declare or cast to string, or use <string> when calling the function; if you just pass in string literals with double quotes the template argument will be const char\* instead of string, and not give you the desired result. | 30% |