Object-Oriented Programming I **Fall 2019**
CIS 9310

Programming Assignment 2

For this assignment you are to write a program that will read in two integer values *x* and *y*, and then uses these values to generate a table of the *square* and *square root* of each integer value between the lower of the two values *x* and *y*, to the higher, inclusively. That is, the two values can be entered in either order. (Note: You may assume that neither value is negative.) You may use the table in Figure 1, which shows the output for an *x* value of 101 and a *y* value of 99, as a guide for how you should structure your output. You may use tabs to separate the columns of your table. Once the table has been output, your program should ask the user if another table should be generated. If the user answers in the affirmative by inputting the letter "Y" in uppercase or lower case, the program should again prompt the user to 2 more integer value *x* and *y*, which is then used to generate another table. This action should continue for as long as the user responds in the affirmative. The program should terminate when the user responds in the negative by inputting the letter "N" in uppercase or lowercase. If a value other than "Y" or "N" (uppercase or lowercase) in input, then the program should continue to prompt the user to input a correct response until one is input.

Your program should incorporate and use the following functions:

double Power (double b, int e);
This function takes a two input parameters **b** (the base) and **e** the exponent) and returns the value b^e. That is, it returns **b** (the first parameter) raised to the power **e** (the second parameter), where **e** can be any (negative, 0, or positive) integer value. (Note: even though a negative exponent may not be used for this assignment, this function should still be able to work for a negative or 0 value for **e**.)

bool MoreData();
This function is used to ask the user if there is more data to be processed, and then processing the user's response. The function returns the value *true* if the user responds in the affirmative, and *false* if the user responds in the negative. If the user inputs a response other than "Y" or "N," that is in invalid response, then this function should continue to prompt the user for a response until a valid response has been input.

double Sqrt (double v);
This function takes a single input parameter, and returns the square root of that value. The square root of a number *N* can be approximated by iterating through the expression

new_value = 0.5(old_value + N / old_value)

To begin the iteration you should start off with an initial value of *1* for *old_value*. Iterating through the above expression will cause *new_value* and *old_value* to converge to the square root of *N*. Because of the way decimal values are typically represented in a computer you may find that *new_value* and *old_value* do not always reach the same value. For this reason you should terminate the iteration when (**the absolute value of**) their difference is less than the value *0.005*.

Output Table:

N	N Squared	Square Root (N)
99	9801	9.95
100	10000	10.00
101	10201	10.05

Figure 1

Test Input Data:

1. 2, 11
2. 103, 95

Due Date: October 22, 2019