# Lab 13

## Templates

Skills Required

* Templates
* Work with arrays and vectors, Create Functions, Include Headers and other files, Loops (while, for), Conditional (if, switch), Datatypes, assignment, etc.

**Assignment**

For this lab, you will write three functions, then convert them to template functions so they may work with any data type.

Template functions work by allowing any data type by just adding a line to the beginning of the function:

template <class T>

T someFunction(T parameter1) {

…

}

T becomes the template type, and so anywhere in this function there is a T, it is replaced by the type that is passed in. So you can call this function in several ways:

someFunction(0) returns an integer

someFunction(1.23) returns a double

someFunction(“String”) returns a string

Since it can identify the parameter’s type of T (which is an int, double, or string), it knows that the function will return the same type T (an int, double, or string).

Incidentally, ‘T’ is just a place-holder; that label can be anything. Sometimes something else is used to indicate some property the data should have:

template<class comparable>

void sort(comparable data[], int array\_size); // must be possible to compare data items

template<class Numeric>

Numeric geometric\_mean(vector<Numeric>& data); // must be a numeric type

// but that type may include Complex, Fraction, or other user-defined type

**Input**

You are provided three input files. The first file, integers.txt, contains 100 integers. The other two files are doubles.txt and strings.txt also contain a list of 100 of doubles and strings (single words).

**Part 1**

Start a Visual Studio Project, and create a file called Lab13Main.cpp. In your main() function, load in the 100 integers into an array. Then, write these functions:

* A void function called mySwap() that takes in an array parameter, the size of the array, and two integer positions, and will swap the items at those positions in that array
* A function called myMin() that takes in an array parameter and the size of the array, and returns an integer that is the lowest number out of that array
* A function called myMax() that takes in an array parameter and the size of the array, and returns an integer that is the highest number out of that array
* A function called mySearch() that takes in an array and its size, and an integer to find, and will return which array position that integer is located at, or -1 if it is not present.

Make sure these functions work before continuing to part 2. You may wish to debug them by printing some information to a file or on screen to verify that they do.

**Part 2**

Load in the doubles and strings files, and store those in arrays. Then, modify those functions to work with any data type by making them template functions. Pay attention to what parameters and return types need to be changed to templates and what do not. (Ex: you can’t have a double or string as a position for an array, for example: myArray[“string”] is not valid. Test the same functions to see if they work with the new types. (For example, mySearch() should work on ints, doubles, and strings.)

**Output**

The output will be a text file named output.txt. In it, you will print out some data that proves the functions work. First, print out the type you are working with. Second, print out elements 10 and 20 of the array, swap them, then print them out again. Third, print out the minimum and maximum of the set, then print out if these items are in the list:

Integers: 1 and 5

Doubles: 4.62557 and 1.23456

Strings: Shoes and Pumpkin

You are given a test file with code that will do this. You will need to copy your template functions into this file to have a working program, but if you have done everything correctly, you should not need to modify anything in the main function of the test program.

Integers:

Swapped items at positions 10 and 20

Before: [10] 78 [20] 81

After: [10] 81 [20] 78

Minimum: 2

Maximum: 99

The number 1 is at position -1

The number 5 is at position 38

Doubles:

Swapped items at positions 10 and 20

Before: [10] 7.69861 [20] 7.21923

After: [10] 7.21923 [20] 7.69861

Minimum: 0.00569

Maximum: 9.9935

The number 4.62557 is at position 19

The number 1.23456 is at position -1

Strings:

Swapped items at positions 10 and 20

Before: [10] Album [20] Vacuum

After: [10] Vacuum [20] Album

Minimum: Airport

Maximum: X-ray

The word Shoes is at position 69

The number Pumpkin is at position -1

**Submit your assignment**

Update things on GitHub or clean up the Debug directory, and upload either the GitHub link or your zipped project folder to the Canvas site.