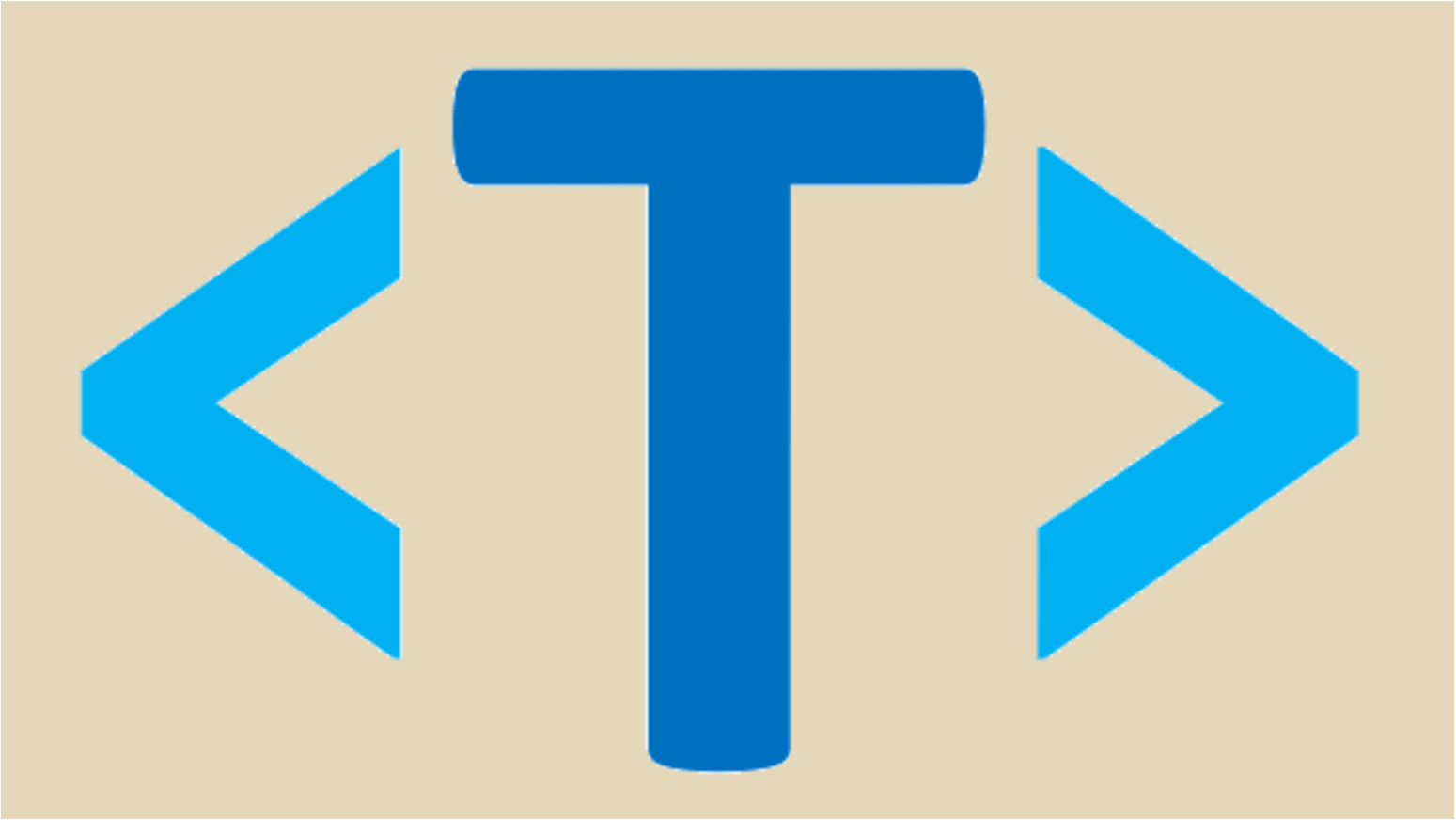
# Lab Assignment 4

**Due** Tuesday by 11:59pm **Points** 19 **Submitting** a file upload **Available** Sep 29 at 8am - Oct 7 at 11:59pm 9 days

## Generic Programming



Make sure you have read and understood

- ***Unit module 5***
- ***C++ Coding Style Guidelines*** **(https://ohlone.instructure.com/courses/18987/files/3097729/download?download_frd=1)**

before submitting this assignment. Hand in only one submission.

# Lab Assignment Objectives

1. Define a function template to be used as a tool for creating a set of functions that have the same code logic but whose code can be applied to different data types.
2. Separate interface and application files for a template function.

# Function Templates

Function templates allow the programmer to define a function whilst deferring the definition of the types until the program is completed. How this works is that when the program is compiled, the compiler creates as many versions of the functions as there are function calls with different types. The syntax included the reserved words ***template*** and ***typename*** in angle brackets. To facilitate code reuse, function templates are placed in an interface file that is included in the program being compiled.

## Understand the Application

Write a templated function to find the index of the smallest element in an array of any type.

## Test Run Requirements:

Test the template function with six arrays: two of type int, two of type double and two of type char. Print the value of the smallest element in each array. Use the sample test cases shown below for part of your test run verification. Be sure to include the additional test cases required to complete your test specification requirements.

## Additional Requirements

- 1) How many versions of the template function does your compiler need to generate at run time to accomplish the specification test run requirement?
- 2) Personalize the nameguard on your header file.
- 3) Submit the lab 4 quiz.

# Lab 4 Quiz

Submit **lab 4** quiz. This is a T/F quiz on generic programming worth 5 points towards your lab 4 grade.

# What to Turn In

Hand in 2 files: **No zip files.**

- **a4.h :** interface file
- **a4.cpp :** application file

Submit your quiz answers.

# Sample Output

Here is an example of a ***partial*** run sample:

```
/* ------------------- Sample run -----------------------------
Smaller of x and A: A
Smaller of 149 and 182: 149
Smaller of 56.7 and 11.3: 11.3
------------------------------------------------------------- */
```

# Tips and Requirements

1. Ensure that your solution is well organized. Provide a program header and comments to document and organize your source code. User defined function(s) need be documented.
2. **Provide a statement in your program header to state how many versions of the template function your compiler needed to generate at run time to accomplish the specification test run requirement. Include the reason for your answer.**
3. Provide a commented out copy of your program run. Enclose the run inside of multi-line comment delimiters so that your program will run in the grader test bed. Place the run after your program source code in the a4.cpp file.
4. Title your submission files in the format ***firstinitalsecondinital***.filename. (Example: **A**nn **O**hlone would submit files **ao**a4.h and **ao**a4.cpp).

# Submission Resources

For more information on how to submit your assignment, please visit:

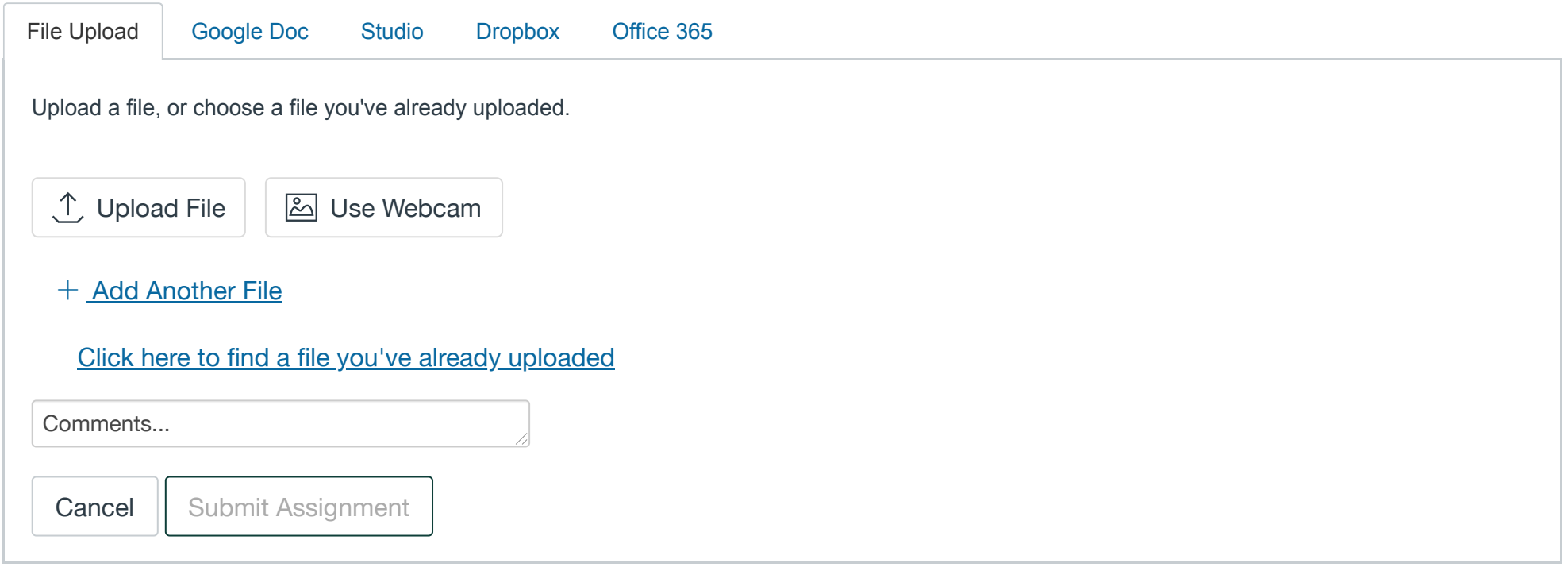
- How do I submit an online assignment? Canvas Student Guide (https://community.canvaslms.com/docs/DOC-9539)
- Assignments Overview Canvas Video Guide (https://community.canvaslms.com/videos/1122-assignments-overview-students)
- Assignments Submissions Canvas Video Guide (https://community.canvaslms.com/videos/1121-assignment-submissions-students)

# Submitting multiple files to an assignment

Your lab 4 assignment requires uploading more than one file; you should upload these 2 files as one submission. In this assignment you need to upload 1 **.cpp** files (a4.cpp) and 1 **.h** file (a4.h). To add these files, the **Add Another File button** is clicked to **upload the two files one by one. Check to make sure that both files uploaded okay.** When finished click **Submit Assignment.**

# Questions?

Feel free to **ask** in the forum!




**Lab 4 Rubric**

| Criteria | Ratings | | | Pts |
|---|---|---|---|---|
| On time submission | **4 pts** **On time** | **2 pts** **One day late** | **0 pts** **Two days late** | 4 pts |
| Header file<br>Satisfies the function template .h requirements : naming guard, template function definition and include file(s). | **4 pts** **Full Marks** | | **0 pts** **No Marks** | 4 pts |
| Driver file<br>a4.cpp : Satisfies the test run validation source statements as specified in the assignment. 3 different data types arguments are demonstrated to include (int, double and char); 6 test cases in all. | **6 pts** **Full Marks** | | **0 pts** **No Marks** | 6 pts |
| Commented out test validation run included<br>A copy of the program test validation run output is attached after the source code in the main.cpp test driver file (enclosed within comment delimiters). The run matches the source code submitted. The test run demonstrates slot machine simulation pulls. | **3 pts** **Full Marks** | | **0 pts** **No Marks** | 3 pts |
| Coding Style<br>Header file includes a program header that describes the template function. Test driver file includes a program header description that describes the call interface; includes answer to the spec question. | **2 pts** **Full Marks** | | **0 pts** **No Marks** | 2 pts |
| | | | | Total Points: 19 |