Assignment Guidelines *(Syllabus Supplemental Document)*

This document illustrates the minimum mandatory assignment procedures, rules and expectation of quality, procedure, and documentation. Please read it carefully and follow it precisely. The document is subject to changes and updates. In case of major changes, students will be notified in class.

1 - Assignment Related Questions:

For questions on assignments, anytime and anywhere, use the online discussion group **(accessible via D2L)**. Students may post and reply to questions in order to help each other. The instructor monitors the discussions and answers questions when appropriate. Use this method so that all students benefit from questions, answers, and clarifications.

If you need help, you can talk to instructor **during office hours**

2 - Due Dates:

A) **Lab (NOT Assignments):**

- Lab work is Pass/Fail and **is your proof of participation**.
- Each failed/not uploaded lab will significantly affect your participation grade *(effect is about 1% to 2% of OVERALL grade per missed/failed lab)*
- Lab work must be uploaded to D2L by end of lab day, no late submission allowed.
- Lab work submitted must reflect acceptable lab work effort in terms of quality and quantity.
- Not to be confused with assignments
- Submitted lab work does not need to follow assignment guidelines.

B) **Assignments:**

- Assignments always have an end-of-day Saturday *(i.e. 11:59PM)* deadline.
- **25% will be deducted per day for late assignments.**
- Assignments must precisely follow guidelines stated below.

C) **Projects:**

- Are due before or on the due date indicated in the project.
- After that date, **there will be %25 deducted per day for late projects.**

NOTE: Remember that projects are long and difficult. They may require research to find out how to solve the problems. For these reasons they have a long due date. Do not wait to the last weekend to attempt working on them as they are specifically designed to be anti-procrastination.

3- Code Style, Including Naming Conventions, Comments, And Marking

Coding style requirements are detailed in the document: **C++ Programming Style Guide**
Failure to follow a consistent, appropriate style **will result in a loss of marks** not only on your assignments, but also on some midterm and final exam problems.

---

4 - Submitting Assignments:

A. Upload your assignments to the appropriate D2L Drop Box

B. Uploaded files MUST be sorted in order of problems. See section C 1.1.2

C. Submitted work must include: **Code (1) & Output (2):**

C-1) **Code:**

1.1.1. Each source code file must be a separate text file (with added .txt extension)
(Due to D2L limitation, you need to rename the file extension and add .txt before submission)

1.1.2. Each file name *(except for classes – see 1.1.2.1)* must clearly contain problem number such that problems are sorted in sequential order. Examples:

Assignment03-1.txt	*A03-P01.txt*	*1- Waterbill.txt*
Assignment03-1.jpg	*A03-P01.jpg*	*1- Waterbill.jpg*
Assignment03-2.txt	*A03-P02.txt*	*2- Waterbill.txt*
Assignment03-2.jpg	*A03-P02.jpg*	*2- Waterbill.jpg*

1.1.2.1. Classes source code: must contain 2 source code files .cpp.txt and .h.txt., files need only to be named the same name as the class contained in them. Each class must have its own .cpp.txt & .h.txt.
Example: *Date.h.txt* and *Date.cpp.txt* for class Date

1.1.3. Each Screen capture file must be a separate .jpg or .png file
Read instructions in section 2 about how to capture and save output window.

1.1.4. ENSURE THAT EACH AND EVERY FILE YOU SUBMIT WILL OPEN DIRECTLY IN D2L.

IF YOUR FILES DO NOT OPEN WITHIN D2L viewer (ie. Requires downloading / does not open within D2L built - viewer) it will not be graded

1.1.5. **HEADER / BLOCK COMMENT:** Every source code file MUST have a block comment at the beginning stating:

> ***your name***, ***StudentID***, *class, assignment description, date of submission and assignment status must be included in the block comment.*

USE this template:

```
/*************************************************************
Problem:      Assignment 2.1
Description:  1 line description of the program/problem. Example: "This program will ask the
user to input number of each student and each student's score then calculates the highest score and
displays it"

Name:  StudentName
ID:    Student ID
Date:  m/d/y

Status:  (complete / wrong results / not compiling / …etc)
*************************************************************/
```

C-2) Sample output for each problem:

Screen capture of output window **(NOT entire screen capture), Image MUST be cropped if needed to show only the output window**:

Using PC:

(Alt-printScreen) will copy active window to clipboard. Then, open "paint" program from the accessories menu and paste the captured image. Finally, save the image file.

Using Mac OSX:

To capture a specific application window, press **Command-Shift-4**, and then press the **Spacebar**. The cursor will change to a camera, and you can move it around the screen. As you move the cursor over an application window, the window will be highlighted. The entire window does not need to be visible for you to capture it. When you have the cursor over a window you want to capture, just click the mouse button and the screen shot will be saved as a PNG file on your desktop. (The file is saved as PDF in Mac OS 10.3 and earlier.)

D. **If an example was given, you must use the same values as the example**

E. **You must submit all your files and samples as one D2L submission, subsequent submissions will overwrite previous ones and delete all files previously submitted.**

F. **You MUST check that your assignment was submitted properly, fully, and opens correctly in D2L after you submit it. Only files on D2L will be accepted.**

5 - Grading Policy:

Assignments are graded for: coding style, quality of solution, usage of things we learned in class, and proper programming practices not only on producing code that works.

All Assignments must follow the rules indicated in this document as well as the style guide document.

Note: Minimum deduction for item is 0.5 points

Accuracy *(including quality of solution, good programming practices, applying what you have learned in class, etc.)* -1% to -100%

Sample items include:

Grade *Min deduction*	Item examples	
100% (i.e. 0 grade)	Plagiarism	If your code has too much similarities with another student's code, taken from another source, or, your output does not match your code. Generally, anything you submit that is not yours. Pls be careful in the future copying/sharing code and also review the academic honesty guidelines in eh course syllabus
	Wrong file format *(per file)*	You must have your source file as .txt which also opens directly within D2L. Double check your submitted files after you submit them.
50%	Block Comment	Your name, StudentID, class, assignment description, date of submission and assignment status must be included in the block comment.
	Wrong solution	You solution does not properly solve the problem.
	Output not submitted per guidelines	Pls follow assignment guidelines precisely. Not submitting output or submitting 1 file for all output images does not adhere to guidelines.
	Global Variables *(Except for const variables)*	Never use global variables except for constants.
35%	Code style	Indentation, clarity, etc.

25%++	Repeating code	You should never repeat code that does the same thing. *For Example: "if" statements should only have a variable that decides on the amount, then everything else (calculation and printing) should be done once at the end.*
	Poor solution	The solution did not need such amount of code.
	No *const* use	You should never use numbers for constant values directly in your code *(example: prices, discount amounts, PI, etc)*. These should be defined as const variables
	Useless checking	For example: when using "else" for range checking you should not recheck again in the following "if" statement.
		Each type of loop has a specific use. Although it is true that you can use them alternatively, this class is about teaching you proper programming.

	Wrong use of loops	Example: the following for loop is unacceptable:
		`for (int i=0; i<10; ){`
		`  .......`
		`  i++`
		`}`

- Note that if you indicate that your assignment is complete and when checked it does not run or produce wrong results, additional points (up to 50%) will be deducted from the grade of your assignment.
- If you indicate incomplete status, you must declare which functionality of the program is not implemented correctly/fully.
- **NOTE**: Once your assignment is graded it can no longer be re-submitted or updated.