24-780 Engineering Computation Problem Set 01

You need to create a ZIP file (It may appear as a compressed folder in Windows) and submit the ZIP file via the 24-780 Canvas course. The file name of the ZIP file must be:

PS01-YourAndrewID.zip

For example, if your Andrew account is *hummingbird@andrew.cmu.edu*, the file name must be:

PS01-hummingbird.zip

If your ZIP file does not comply with this naming rule, you will automatically lose 5% credit from this assignment. If we are not able to identify who submitted the file, you will lose another 5% credit. If we finally are not able to connect you and the submitted ZIP file, you will receive 0 point for this assignment. Therefore, please make sure you strictly adhere to this naming rule before submitting a file.

The ZIP file needs to be submitted to the 24-780 Canvas course. If you find a mistake in the previous submission, you can re-submit the ZIP file with no penalty as long as it is before the submission deadline.

Notice that the grade will be given to the final submission only. If you submit multiple files, the earlier version will be discarded. Therefore, if you re-submit a ZIP file, the ZIP file MUST include all the required files. Also, if your final version is submitted after the submission deadline, late-submission policy will be applied no matter how early your earlier version was submitted.

Make sure your program can be compiled with no error in one of the compiler servers. Don't wait until the last minute. Compiler servers may get very busy minutes before the submission deadline!

Submission Due: Please see Canvas.

PS1-1 Download and read the course syllabus from the Canvas course. (15 pts)

Make sure you can log on to the 24-780 Canvas. Download and read the syllabus. You do not have to submit anything for this assignment.

PS1-2 Download and install a developing environment. (15 pts)

Follow the instructions on the lecture note and install Visual Studio if you are using Windows, or XCode if you are using macOS. You can work on assignments on Linux using clang++, in which case you will need to study how to set up and use clang++ by yourself, or you can come to the instructor office hour for questions.

PS1-3 Correct Errors in the Count-Down Timer [ps3-1.cpp] (70 pts)

The following code is supposed to take the number of seconds as input and count down until it reaches zero. When it reaches zero, it prints "Time is up." But, it includes a bunch of errors. Fix all errors and make it work.

Pay attention to the compiler error messages. If it says missing semi-colon, then a semi-colon is missing somewhere. Or, if it says newline in constant, it means constant text string is not closed somewhere.

It is recommended to remove all warnings as well, but it is not required for this assignment.

(The following two paragraphs were missing when I uploaded the document earlier. Sorry)

Save the C++ program as ps1-3.cpp and submit a Zip file that includes the source code (ps1-3.cpp) to the 24-780 Canvas course.

Make sure your code can be compiled by Clang and Visual C++ with no error by one of the compile servers before submission.

Good luck!

To Answer common questions I had in class:

The count-down output may or may not start the number you entered. And, you may see 0 in the end or can be 1. But, don't worry about it. It is ok as long as it counts roughly the number of seconds that the user enters.

```
#include <stdio.h>
#include <time.h

// If you are more comfortable with std::cout / std::cin,
// you can replace C-standard library functions with std::cin and std::cout.

int main(void)
{
    printf("24-780 Engineering Computation Problem Set 1-3\n");

    printf("Enter how many seconds to count:");
    int t;
    scanf("%d",&t)

    int tEnd=time(NULL)+t;
    int prev=time(NULL);
    while(time(NULL)<tEnd)
    {
        if(prev!=time(NULL))
        {
            printf("%d\n",tEnd-time(NULL));
            prev=time(NULL);
    }

    printf("Time is up.\n);

    return 0;
}
```