CS2024 Commenting Guidelines

Commenting in any computer language has historically been a contentious subject! Arguments about how necessary commenting is go back to even before I was learning how to program back in the 1980s! Some will say that if you choose descriptive variable, function and class names that what your code is doing should be "intuitively obvious" and commenting will be redundant. Others will insist that the definition of "intuitively obvious" is very subjective and what might be obvious to you may not be to others.

This brings us to the point of commenting. I believe there are two main reasons that we comment code. The first is for any other programmers that may need to maintain/modify code we write. It's a chance to explain to them what you did and why so that they have as good of an understanding of your code as possible before they modify it. The second reason is to help you remember what you did and why "down the road". Down the road may be 6 months from now or two (or more?) years from now. What is perfectly clear to you today may not be if you come back to this code years later!

However, beyond your personal preference for how much commenting you think you should do for both the reasons mentioned above, the reality is that if you work on a team of programmers in an organization, that organization likely has *coding guidelines* that would include how you should comment your code. In that spirit, I'm going to provide you with *commenting guidelines* for CS2024. If you do not follow them, you can expect to have points deducted from your assignments.

There should be a file named `sample.cpp` linked from the main page of the course web site which exemplifies our coding guidelines.

Header Comments

Every file you submit should start with a comment that identifies the filename, the Assignment it pertains to, and a brief description (one or two sentences) about what the code contained in the file does. Consider the following example from sample.cpp:

```
//
//  main.cpp
//  Assignment #0
//
//  This code implements a check to see if a user is old enough to vote
//
```

In addition to the header comments, you should have a comment before every function and class member function defined that gives a quick summary of what the function does. The only exception is the main() function.

In the body of your functions/member functions, you should comment lines of code that aren't *extremely obvious* as to what they do and why. The code in `sample.cpp` may err on the side of caution with respect to this (I perhaps don't need a comment that says "Read in the age"), but it gives you a general idea of what I'm looking for.