CS170 Week 4 Lecture 2 Notes

Disclaimer: These notes are meant to be read in parallel with Professor Mead's online notes if you have missed class. Topics will not be covered extensively. Here, you will only see the minor details Prof. Mead spoke about which were not on his topic notes.

Classes Continued

const Member Functions:
const- It's the most underused keyword.
Gettor functions should always be const because they only read data.
const is also a signature, you can use it for overloaded functions.
You can't have const global functions.

Separating the Interface from the Implementation:
The interface is typically in the public.
We'll put definitions in header files, implementations in the cpp files.

Default Constructors and Destructors:
Maybe we want to allow the class or struct to initialize values.
Global data is guaranteed to be initialized to zero.
For user defined types, the default values must be specified.
"*Point pt1();*" is now considered a function prototype.
This would be a function called pt1 that takes no parameters and returns a point.
We can't do this ^ when it comes to default constructors.
Default doesn't mean the constructor doesn't require any arguments, it means that there are no required arguments from the user.
The only way to get garbage into the class is if you put it there into the constructor.
In fact, in C++, every time you create an object, a constructor will be called.
If you don't make one, the compiler makes one for you. This is for the C backwards compatibility.
If all of your constructors are private, nothing outside of the class can construct an object.
Again, default arguments must be defaulted right to left.
Just like with constructors, if you don't provide a destructor- the compiler will provide one for you.
When we have object oriented file opening and things go out of scope, files will be closed for you.

Lab #4 Info / Post Class Mentions

- ➔ The next four labs will be built off of this one
- ➔ You won't get away with any sketchy doxygen activity on this lab