

CSCI 1301 – Lab 14

October 2, 2018

1 Milestone #2

Since Milestone #1 (Lab 06), you learned a lot. Hopefully, you learned how to write your own study guide thanks to my definitive study guide¹ (printable version²), and you should note that *most of our faculty members in the School of Computer & Cyber Sciences do not provide study guides.*

If I were to write one, I would probably write something like the following...

1.1 Academic Life

- You now hopefully have a clear picture what this class is about, and what a CS / IT degree can bring you and asks from you.
- Not having all the time straight A's as you (maybe) used to do in High-School is common in College.
- That, indeed, this class was constantly re-using what you had studied before, and constructing on top of what you studied the week before.

1.2 The Concept of Class

- That the implementation of a class was taking place in a different file, that uses the `class` keyword,
- What a member of a class is, that attributes represent what an object *is*, and methods what it *does* (or how it can be acted upon),
- That, in all the scenario that we looked at, attributes were private, so that application programs would have to use methods to access them (to *get* or *set* their values),
- How an object could be created, or instantiated, and that every object had their own instance variables,
- How UML diagrams were a useful abstraction to design classes.

1.3 Scopes and Conventions

- That a variable was accessible at a particular time and place,
- That renaming had to be uniform in the scope of a variable or a method name,
- That multiple conventions existed regarding the naming of attributes and methods,
- That attributes had default values, that could be changed if needed.

1.4 Methods

- The difference between arguments and parameters,
- The importance of constructors, the difference between default and custom constructors, and how to write your own,
- The definition of the signature of a method,

¹http://spots.augusta.edu/caubert/teaching/general/study_guide/

²spots.augusta.edu/caubert/teaching/general/study_guide/index.pdf

- What overloading means,
- The difference between static and non-static methods,
- The importance of the `ToString` method,
- That a method can call other methods.

1.5 Various

- How and when to use constant values,
- How and when to use static attributes,
- How to use and navigate in the `Math` class,
- What was the benefits of using format specifiers.

1.6 Using Softwares

- How to have 2 `.cs` files open in one project in VS,
- How to “undo” things, using `CTRL + z`,
- How to use a software to compare text files