Programming Practices and Strategies

When given a programming assignment, it's very tempting to sit down, immediately start writing a bunch of code, and then test it at the end. However, this is very poor practice. Your program will likely end up with a lot of bugs as a result, and you'll end up with a lot of frustration.

The following two videos provide a brief overview of some of the practices and strategies that programmers are expected to follow in industry. They are geared for audiences who are or who are planning to be professional programmers. As this is an introductory CS course, you won't be expected to adhere to all of these practices (ex. gathering user requirements, creating functional specifications, refactoring), but they should give you an idea of what professionals have to do.

In addition, some of these practices will help you in this course. In particular, pay attention to the following:

- Break your mountains into molehills. Don't try to solve the entire problem at once!
- Focus on method inputs and outputs first without worrying about implementation details (black-box programming).
- Use pseudocode so you don't get bogged down by the details of Java syntax.
- The Tale of the Pyramids. An incomplete program will get you a better score than one that crashes and breaks the auto-grader.
- The importance of commenting. We'll talk about Javadocs this term.
- Proper indentation. This will make your code easier to read.

We'll also talk about the importance of testing and of writing your tests *before* you write the methods that are being tested.

Video: Program Design (https://video.wpi.edu/Watch/Pf65WmDd)

Video: Coding Practices (https://video.wpi.edu/Watch/m2E7KyYx)