

CS 251 - Lab 004

TA: Kage Weiss

Office Hours: [TBD] FEC, or by appointment.

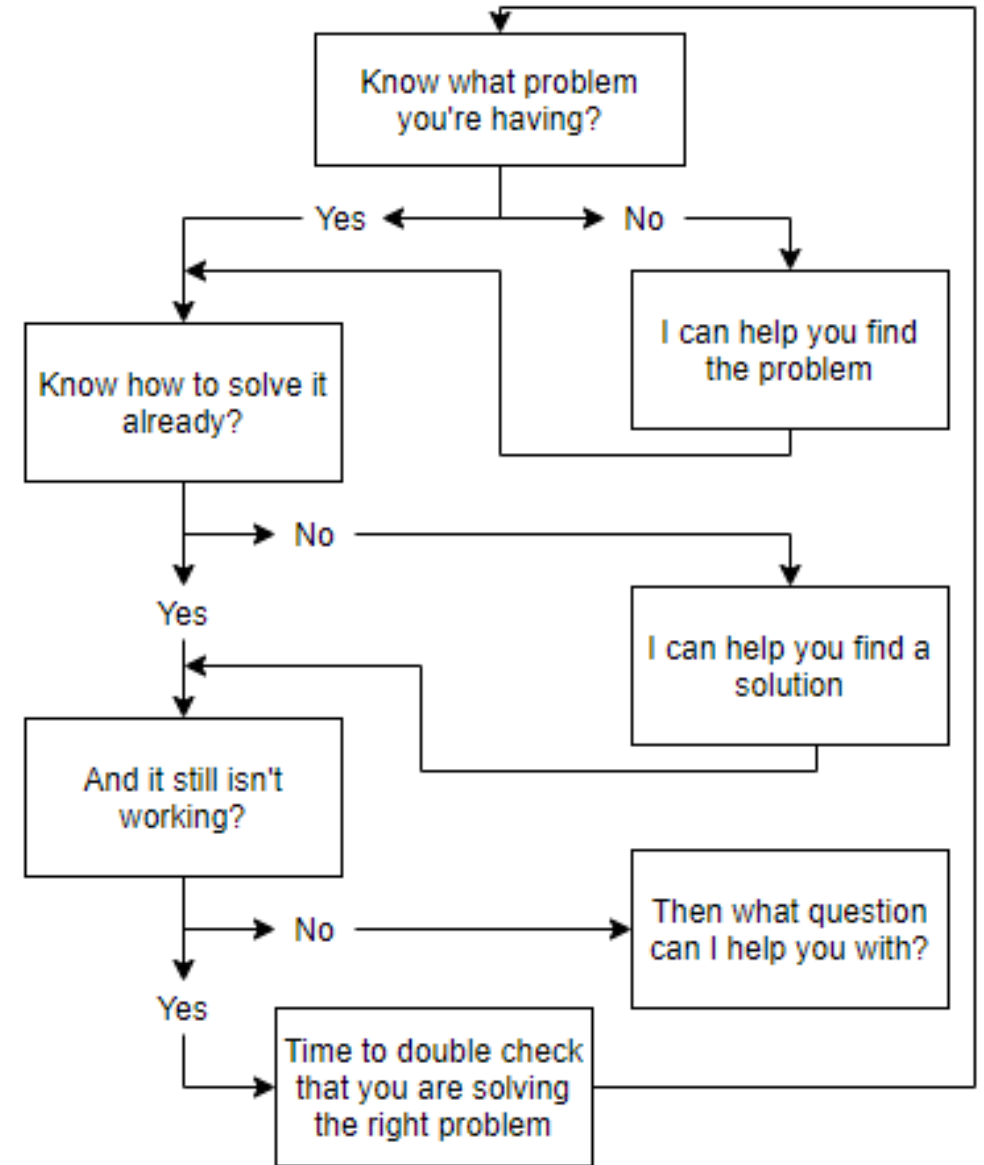
Contact: mmweiss@unm.edu

Website: <http://cs.unm.edu/~kageweiss/TA/cs251.html> -- **SLIDES POSTED**

- Sign in by email today: include your name the keyword from the board, and if you haven't set a profile picture on LEARN, you should do so.
- Today we are working on ensuring you have the proper programs (see below), and FizzBuzz (Lab 1)
 - **Do you have an editor?**
 - **IDEs are okay, but we want you to get used to using a simpler editor (emacs, Notepad++), it'll help when exams roll around.**
 - **Do you have the Java JDK 12? Installed? "javac" works?**

How I Run a Lab:

- Sign in! Guests too! My slides will say whether to sign in by email or on a sheet, this helps show that you are putting in the work next time you need an extension from the professor.
- If the lab we are working on is one that may benefit from some explaining, I'll give a mini-lecture at the start of lab. You are free to ignore me and just get to work if you feel confident in your understanding!
- I'll walk around or be at the desk if you have questions, but please raise your hand if you need help. That's literally why I'm here, so please let me know if you are having problems or questions!



How I Grade Your Code:

- Chenoweth provides the rubric and occasionally tester code.
- I write tester code that thoroughly tests your code.
- If your code runs as expected with my tester I'm happy.
- I then review your code (.java) to make sure it follows CS251 Code Standards.
 - That means variable and method names, comments, privacy, **NO TABS**, etc.
- I **can** tell how much work you put into your code.
- I can tell who didn't cite StackOverflow.
- If you obviously put work in I'm happy to give points and comments!
- If you obviously didn't put work in I have to really look at your code and end up finding (and counting off for) minor infractions.
- Every point I take off is listed in the comments for the grade. **Read them.**

Command Line Arguments

Command line args are an integral part of this course, so we need to check some things:

- Make sure you have the JDK installed and added to your path.
 - Need to check? Open cmd or terminal and type “javac” and hit enter. No errors: you have it in the right place, otherwise google is your friend.
- Google is your friend. Chances are someone on the internet has a whole list of useful shell commands.
- Running from an IDE is fine for testing, but if you don't also compile and run from the command line, I can't guarantee your code will work when I compile and run it on the command line.

```
1  /**
2   * @version date (in_CS_XXX_00X format : YYYY-MM-DD)
3   * @author FirstName LastName
4   */
5
6  /** My class ClassName does ... */
7  public class ClassName {
8      ....
9      private memVarType memberVariable;
10     ....
11     /**
12     * Getter for specified member variable.
13     * @return this.memberVariable The memberVariable of this instance
14     */
15     public memVarType methodName() {
16         return this.memberVariable;
17     }
18     /**
19     * What does this method do?
20     * @param param1 What is this parameter?
21     * @return What are we returning?
22     * @throws ExceptionName Why are we throwing this/what triggers this?
23     */
24     public varType methodName2 (memVarType param1) throws ExceptionName {
25         if (param1 == this.memberVariable) {
26             for (int i = 0; i <= param1; i++) {
27                 System.out.println(i + ". Our var = " + this.memberVariable)
28             }
29             throw new ExceptionName (ExceptionParameters);
30         }
31         return (this.memberVariable + param1);
32     }
33 }
```

} Info Block, tells who/when/what

} Method is public, so it gets a JavaDoc comment

} Method is public, so it gets a JavaDoc comment, this one's a bit longer because it has three fields

} Indentations must be spaces, NOT Tabs