

## Student Success Guide

CSE 143 is a hard course. A lot of concepts won't make sense at first, but we try to provide as much assistance and resources as possible to get everyone to the point where they can succeed. Our goal is for you to feel confident using the skills you learned for whatever programming they might take on in the future. Here we have documented a few things the TA's have felt is imperative for success in CSE 143.

### General Tips

- Attend section
  - There really isn't a way to improve your programming skills without practicing programming
  - You could practice on your own, but your TA's will guide you through problems that have key insights into learning new concepts
  - It's also a place where TA's will discuss common style mistakes that occur in new programming concepts
  - It's ALSO a place where you can tease out your misunderstandings of concepts by discussing with your classmates
- Attend/watch lectures
  - When you get to attending section, it's a lot more effective practice if you have a foundation to work off of
  - Lectures are made so you can sit back and listen and let the concepts first be introduced
  - We also go over common mistakes in programming here too!
- Reach out to a TA whenever you're confused/struggling with ***LITERALLY ANYTHING***
  - TA's are almost always willing to help out whenever you might be confused about something
  - If you have questions about a certain general topic, we can go over section and practice materials in great depth to get you caught up, but we can't go in-depth on the homeworks themselves
  - Even if you have general questions such as time or stress management and the like, they can offer what advice they can

### Assignment Specific Tips

More specifically on the assignments, we have put out a lot of resources for you to refer to. It can be a little overwhelming at first however so we'll describe each of the resources below and make a checklist for how to go about working with these resources.

[Assignment Specifications](#): The specifications detail all the methods that your class should have, as well as significant guidance on how you should be implementing your class. It also contains some limitations on the implementation and style issues for you to consider as you implement it. There is also a useful FAQ section for each homework that you can read through.

[Output Comparison Tool](#): The Output Comparison Tool will have some sample logs that you should use along with the given client program(s) to test if your object is behaving correctly. While matching all the logs on the Output Comparison Tool does not guarantee full points on external for your assignments, if you don't pass all the logs, you are guaranteed to not get full points on the assignment. It is up to you to fully test your code on any other possible edge cases that might come up.

[143 Style Guide](#): The style guide is a fairly comprehensive document on all the style mistakes that students commonly make when learning Java. This is a go-to resource for whenever there are certain things that you

are confused about. This is very large though, so you shouldn't read it all in one sitting, but refer back to it when you have questions.

[143 Commenting Guide](#): The commenting guide is an introductory guide for how to document methods and classes in CSE 143. You should read this before your first assignment and refer back to it throughout the quarter

[General Style Deductions](#): The General Style Deduction highlights common mistakes that students make regarding each assignment. It also has a list of Forbidden Features that students are disallowed for using in this course. Like the Output Comparison Tool, it doesn't highlight every single possible mistake that students make, so following it won't guarantee full points, but you cannot receive a full score on your internal code style without making sure you attend to each point in the General Style Deductions.

### Checklist

When the assignment is first released:

- Read through the assignment spec and all the General Style Deductions (past assignments too!)
  - Ask a TA in office hours or on the message board if you have any questions about the coding style concepts

When working on the assignment:

- Write "method stubs" for each of your public methods, where you create the method headers for each method required in the spec and comment the methods.
  - If you comment the methods before implementing the internals of any method, you're a lot less likely to include implementation details!
- Work through the homework with the suggested development strategy in the spec
  - Test your code incrementally, eg. Just make the constructor and test that it works properly with your own client program that you create, then implement one method and test that out, etc.
- Go to office hours if you need help and get stuck!

When you feel finished:

- Check the external behavior of your program by making sure you pass all the logs of the Output Comparison Tool
- Test that your object will throw the correct exceptions in the correct situations with your own client program
- Go back through your program and make sure all the comments are sufficient. Do they fully document how to use your object for a client who has never used it before (without revealing implementation details)
- Go back through the assignment spec and General Style Deductions to make sure you didn't make any common mistakes!

Once you receive feedback for the assignment:

- Jot down all the mistakes that you made in an on-going list of mistakes
  - Make sure you didn't make the same mistakes on the assignment you are about to turn in. Go through the entire list so you don't make the same mistake you made before!