

CSE 143: Computer Programming II

Tips/Resources for Success in 143

This document contains tips and information about the many resources we provide for you to get help in the class when you need it. Maybe you have forgotten some of these after taking 142 or maybe you didn't take 142 at all, but all of these resources can be helpful for you!

(1) Course Staff

The course staff is here to help you! We are always happy to help explain concepts or answer questions about the course. Here are some ways that you can get help from a member of the course staff:

- **Email:** see the [Course Staff Page](#) for email addresses.
- **Message Board:** We regularly check this and try to answer questions promptly
- **Hunter's Office Hours:** Mon 12:00pm - 1:30pm, Wed 1:30pm - 3:00pm, or by appointment in CSE 218
- **IPL:** The TAs staff our Introductory Programming Lab (IPL) nearly every day of the week (see the [schedule](#))

(2) IPL

Here are some good ways of using the IPL

- We highly recommend having a TA teach you how to use the debugger tool in your IDE here. It is invaluable when we start dealing with more complex data structures like linked lists and binary trees. One of the largest keys to mastering the class material is being able to visualize all the data structures, and jGRASP's debugging tool is prime for this.
- If you find a section problem related to the homework material that you are unable to figure out on your own, this is a great thing to work with a TA on in great detail.
- Asking the TA to give you a mini-lecture on a concept or example you didn't quite get from lecture/section. This is another area where TAs will talk as in depth as they want about the material.
- Here are a couple examples of good IPL Questions:
 - "My code isn't working. Can you show me how to use the debugger so I can figure this out?"
 - "I was debugging my code and for some reason even though I returned the variable from my helper method, the value wasn't updated in `main`. Do you know why that is?"
 - "I don't understand what Hunter meant when he said `X` is bad style. Can you explain?"

Here are some bad ways of using the IPL

- Expecting the TA to point out exactly which lines of code are broken and/or how. The TAs are happy to show you debugging strategies and help you step through your code, but you should be deciding what changes to make.
- Asking a TA whether or not something will lose style points. TAs will not pre-grade your assignment. You can ask general style questions and discuss pros and cons of a certain approach, but how you solve the problem is ultimately your decision.
- Here are a couple examples of bad IPL Questions:
 - "My code is broken. Please tell me the line of code I have to change to make it work."
 - "Will doing `X` on my assignment lose style points?"

(3) Message Board

This is where you can ask more specific questions about the homework spec or when the IPL is closed. Other students, TAs, and Hunter monitor this board so you can potentially get help more quickly this way. **Please read the [Message Board Policy](#) before using the message board.**

(4) Guides for Homework

On the [Homework page](#), there are links to many useful style guides and other resources to help you on your homework. Some things to note: [Style Guidelines](#), [Commenting Guidelines](#), [New General Style Deductions](#) (make sure to read top of the page thoroughly)

(5) Tools

There are various tools provided to help you succeed in the course. We find that the students who succeed in 143 are the ones that use these tools regularly throughout the quarter.

Output Comparison Tool

The [Output Comparison Tool](#) is where you can check if the output of your program is performing as expected. Because we provide this tool, we expect that your output matches exactly with ours. Many students lose external points because their output doesn't exactly match ours on the cases we provide.

Note: Getting perfect matches on the output comparison tool doesn't guarantee any particular number of points on the homework, but it is a really good signal that you are on the right track.

Practicelt!

[Practicelt!](#) is an archive of a couple thousand practice problems and exam problems so you can review concepts and make sure you have them down. You get immediate feedback, which is really nice. Hunter will post links on the course calendar to [Practicelt!](#) problems he thinks are a good way for you to gauge if you understand the material from that lecture.

Warning: Around exam times, [Practicelt!](#) tends to crash a lot because everyone is frantically practicing and overloading the servers. Try to use [Practicelt!](#) regularly through the quarter to help master the material as you learn it so you don't need to rely on it too much when the exam rolls around.

Indenter Tool

The [Indenter Tool](#) can fix your indentation. You copy and paste code bits in and it will fix the indentation, then you can copy and paste the code back into your program. If you are using Eclipse, this tool is not useful as Eclipse can auto-format your code for you.

Warning: This doesn't work perfectly if you omit curly braces around single line if and for loop statements. Use with caution and make sure to read the disclaimer on the page.

Special thanks to Miri Hyman for creating most of the content for this document