FAST FACTS

How to Successfully Manage CS Programming

Learning how to program is like learning a new language: you have to learn the syntax, the grammar rules, the common format and styles and, most of all, you have to practice, practice, practice.

What follows below are seven study skills tips for those taking computer science courses, whether you're taking introductory programming or you're taking higher level CS classes.

Lesson 1: Do your own programs.
You probably know a computer science major willing to do your program in five minutes. Alternatively, maybe that person will not do your program for you, but is willing to tell you exactly what data structure you need to build in order to solve the homework problem. **Do not fall into this trap**. Sure, it might save you a few hours of frustration, but you will not learn if you are always dependent on someone else to solve your problems for you. In the end, you are only cheating yourself. You are smart enough, and the struggle that you go through figuring out your homework will always be worth it in the end. In fact, most computer science professors test their students on problems similar to the homework problems, but slightly different, in order to see if you actually did your own homework or if you outsourced it to someone else.

Lesson 2:
Understand the theoretical concepts before you begin programming.
You will save yourself a lot of time debugging code that you did not understand in the first place if you work on understanding all the classroom concepts before you begin programming. The programs assigned will always test your understanding of the concepts from your books and lecture. If you don't understand these concepts, it is unlikely you will have the necessary skills to complete the homework assignments. Don't waste your time: study before you program. The alternative: **code in haste, debug forever.**

Lesson 3: To get there, you've got to know where you're going…
Look for the big picture! Generally, it is better to lay out a plan of which pieces of code you need to write, and then implement and test them one at a time. Doing this will help keep in perspective what needs to be written. Also, if you test the code as you go, you may find the annoying bugs sooner. Plus, it will be easier to schedule yourself and set very measurable milestones. Finally, comment your code as you write it, so that when you come back to it later, you easily remember what you did. This may seem like a hassle at first, but it can save you a lot of trouble and time down the road.

—over—

Lesson 4: Go to class.
Class is important. You might think that downloading the PowerPoint from BlackBoard and reading the book is comparable, but often professors integrate the material differently in class. In addition, class time is a great time to ask questions about the material or about the homework assignments. Examples done in class are always better understood when you're sitting in class listening to the professor than when you're reading a PowerPoint at 3 a.m. the night before the exam.

Lesson 5:
Ask for help when you need it.
Do not be afraid to ask for help. Office hours are a great time to meet with TAs or professors to better understand the classroom concepts, to clarify the requirements of the homework, or to get some help debugging code. In addition, Academic Development provides free tutoring for a variety of the programming classes offered at Carnegie Mellon.

Lesson 6: Practice often.
Just like a foreign language, if you don't practice often, it's easy for you to forget what you've learned. For those new to programming, make sure you keep on track with the lectures and try to practice programming as often as possible. For concepts that you are unsure about, asking for additional examples in office hours or during class can help greatly.

The Learning Zone
Carnegie Mellon
ACADEMIC DEVELOPMENT
412/268-6878
Cyert Hall, Suite B5
www.cmu.edu/academic-development