# Suggestions for doing programming assignments

Too often, we get emails like the following;

> I got started late on the assignment and underestimated the difficulty. Can I have an extension?
>
> I have an exam on the assignment due date, and I have to study for it. Can I get an extension?
>
> I thought the assignment was due tomorrow, not today. Is it better to hand in something unfinished or finish it and take the late penalty?

These are often due to time-management issues. You have to look ahead and plan your time. This takes discipline.

There are essentially two entirely different ways to approach a programing assignment. We discuss them below.

## 1. Procrastinate[1]

Wait until the day before an assignment is due to start on it. Here's what happens. You don't understand something and go to office hours to get help, but 10 other people got there before you, and you don't have time to wait. You rush to the consultant office but no one is there —you forgot to check the pinned office-hour note on the Piazza. You are getting worried.

You finally talk to someone and get the answer. You code furiously in the few hours you have between all the other things you have to do. As you code, you know you have errors, but you want to get everything finished before looking for errors. You know you are not following a sound programming methodology, but you don't have time for all that methodology stuff and just want to get it done.

At 6PM on the due date —you have to submit before midnight— you finally start testing and debugging. But there are errors and you can't understand the output you get. You spend an hour studying the code, patching, running again, to no avail. You head for the consultant office. It's a zoo, a gymnasium, with 30 people in the same boat, sweating and worrying, clamoring to talk to a consultant. You wait, very impatiently, looking over your code and commiserating with everyone else waiting.

Finally, you talk to a consultant. You want them to tell you what your errors are. But that is not their job, they tell you! Their job is to help *you* debug, to give you suggestions for putting well annotated println statements in the program, for example. But you don't have *time* for that. You're frustrated, angry.

This is not a good learning experience! You aren't learning anything! You end up tired, frustrated, and mad at yourself and the course, having learned nothing.

## 2. Start early, work steadily, finish early

An assignment will be a good learning tool if you start early, work steadily, and finish early. Perhaps you work on it every other day for an hour. You aim to finish a day or two ahead of the deadline.

First, read the whole assignment handout, not understanding everything but getting a good sense of what the assignment is about. Then, lay out a timetable (if a suggested timetable is not given with the assignment) for completing the various parts, aiming to finish ahead of time. Then, stick to that timetable!

The benefits should be obvious. If you run into a problem, you have time to do some reading, to talk to friends, to get help from a staff member, to check the Piazza assignment post for answers, and to ask a question on the Piazza. Also, when working on the assignment every other day, you will be pondering it subconsciously and consciously in between everything else you do, and that is good.

Working in this fashion, the assignment becomes the more leisurely, fun, learning tool it is supposed to be. You will enjoy it so much more and have so much more success. With a lot less stress.

[1] How's this: Whenever I start a new task, I procrastinate immediately so that I have more time to catch up.