

Guide to FYS 187-4 Success

My desire as a teacher is to provide great value in your course experience and the tools necessary to success both here and beyond. Here are some of the tips that experience has shown are most important in student success:

- **Grit:** “[How Children Succeed](#)” by Paul Tough makes the case that *grit* (i.e. perseverance, determination, tenacity) is one of the characteristics that are most highly correlated with success in life. While I encourage students to seek help in office hours when significantly blocked in their progress, I also encourage development of grit in
 - Reading carefully – Don’t settle for partial comprehension; go for mastery. When asking questions, ask good, focused questions. The statement “I don’t understand any of this” is never true in my experience. Start with what you understand and ask yourself specifically.
 - Problem solving/implementing patiently – Never expect success on the first try. If you do succeed, celebrate; if you don’t, it’s par for the course. The discipline of diagnosing problems is an important part of your growing skill set. Be gracious towards yourself as you work through your normal human imperfections towards a perfect solution.
- **Class participation:** Woody Allen claimed “80% of success is showing up.” Maybe 80% is too high, but being here physically and mentally is essential. There is much that will be offered in class that will not be available through readings. As an undergrad, I despised what I termed “textbook courses”, that never went beyond textbook material. Thus, you will learn material beyond our readings, and I will make it a point of providing unique examples in class that differ from those of our readings.
- **Readings:** (See <http://cs.gettysburg.edu/~tneller/fys-187-4>.) Our readings are like another teacher presenting another perspective, different examples, etc. If you only read what you must in order to do the sampling of exercises, you miss topics. Imagine the cumulative knowledge as a “Jenga” tower, with each chapter being a layer. Sampling readings for assignment answers is like building a “Jenga” tower with missing blocks.
- **Disciplined assignment work:** This is most important.
 - **“One must learn by doing the thing; for though you think you know it, you have no certainty, until you try.” – Sophocles; “We can only possess what we experience. Truth to be understood must be lived.” – Charlie Peacock**
Assignment exercises are where you take what you think you know, deepen your knowledge, and gain certainty of mastery.
 - **View it as a mental exercise program you take seriously.** Just as a physical trainer prescribes an exercise program to stress your body the right amount in order to increase fitness, I act as a “mental trainer” to assign exercises that stretch you to increased problem solving capability. Just as a week’s worth of workout hours are best divided up with time for muscles to repair/grow between sessions, your homework is best spread out across multiple sessions throughout the week. Just as trying to do all hours of week’s workout at once is likely to lead to injury, trying to do all of a week’s homework

is likely to lead to disappointment, frustration, demotivation, etc. On the other hand, *success leads to further success.*

- **Best practice for programming assignments:**
 - Read the assignment ASAP after it is assigned. Do *not* program immediately. Read the problems and think well about a structured approach while in the shower, walking from point A to point B, etc. You'll overcome obstacles subconsciously and save yourself from false starts. Don't code without a plan.
 - Start early. Do begin after a couple days. Things usually take more time and/or effort than we first estimate. Aim to finish days before the due date. Best case: early finish, no stress, happy-happy-joy-joy. Worst case: ample time to work through "bugs", seek assistance if needed, etc.
 - Test, test, test. A program that works for a single test case doesn't necessarily work for all test cases. You can't cover all test cases. Choose enough test cases to exercise all lines of code, boundary conditions, etc. Remember, this is one of the few subjects where you can guarantee an "A" grade through quality work.