Guide to CS1 Success

Programming is to CS as writing is to English. It's the medium for our art. To become a *good* programmer, you need **good examples** (attentive participation in class, readings) and **much exercise** (disciplined assignment work).

- **Class participation**: "80% of success is showing up." Maybe 80% is too high, but being here physically and mentally is essential. This is not merely a "textbook course", as you'll learn material beyond the text, and I make it a point of providing unique examples in class that differ from the text's.
- **Readings**: (See http://cs.gettysburg.edu/~tneller/cs### ← Insert your course number here.) Our text is 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 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:**
        - 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.