CSCE 110 — Programming I
Final Remarks

Dr. Tiffani L. Williams

Department of Computer Science and Engineering
Texas A&M University

Fall 2013
Final exam information

- Final exam review: Tuesday, Dec. 3rd during regular class period.
- Final exam: Tuesday, Dec. 10th from 1 to 3pm.
- Final exam study material will be available on elearning after the Thanksgiving holiday.
- Complete the final survey about programming for 5 points added to your final exam score. The final survey is available in elearning.
- Also, if you have not filled out the class survey administered by Texas A&M, please do so at your earliest convenience. Go to http://pica.tamu.edu to fill out the class survey. It should only take a few minutes.
What a great journey we have taken! (1)

1. We learned the basics of programming in Python.
   - Variables and Expressions
   - Decision-making and repetition
   - Collective structures: strings, lists, tuples, sets, dictionaries
   - Functions and modules

2. We learned how to use the above basics to solve a bunch of problems using Python. Here are a few examples.
   - **Games**: Guessing Game, Blackjack Dice, Craps, Lingo
   - **Simulation/Modeling**: coin problems, dice problems,
   - **Text Processing**: CSV files, Line/character/word counts
   - **Data Visualization**: Plotting different kinds of data with matplotlib (e.g., Wing Bowl, Cost of Missing Class)
What a great journey we have taken! (2)

- You have written a lot of Python programs this semester. How many programs do you think you have written?
- Sometimes you were asked to write programs that were straight-forward.
- Other times, you were asked to take lemons and make lemonade — or at least that’s probably what it felt like.
- Each of you took ownership of your thoughts to solve the problem.
  - Solving a problem starts with confidence and accepting that you might have a few false starts before the right path is found.
  - Each solved problem gives us the confidence to accept greater challenges.
  - Programming is a great tool for expressing yourself creatively to solve a problem.
  - It’s amazing to see how 100+ minds think differently to solve the same problem!
What is programming?

- At the beginning of the semester, I said that programming is the procedure for taking input and transforming it to output. While this is true, this definition in many ways is unsatisfactory.
- Programming is mathematical or computational thinking.
- Benefits of programming include:
  - Learning the discipline of serious thinking.
  - Learning to take pride in your work.
  - Learning how to solve a problem is empowering.
  - If you can program a task, you most likely understand it. The computer is the ultimate teacher.
The future of programming

- Will every person need to know how to program in order to be considered educated?
  - Can’t the people who need programming just buy it or download it for free?
  - Possibly. But, think about how people in the Middle Ages or ancient Egypt communicated their thoughts. Back then, you hired a better-educated person (i.e., a scribe) who knew the writing language and you needed someone on the other end who knew how to decode it.

- Will the need for a separate scribe tribe of programmers continue through this century? Or, will the skill set of an educated person include programming fluency?
  - No one really knows. But there are many opinions and ideas on the subject.
  - Each of you will play a crucial role in answering this question.
Let’s take a look at the learning objectives of the course as specified in the syllabus.

1. Develop a basic understanding of programming and the Python programming language.
2. See the value of programming in a variety of different disciplines — especially as it relates to your other college courses.
3. Appreciate the value of experimentation.
4. Be comfortable with the fact that there is more than one right solution to a problem.
5. Have fun!

I hope you feel satisfied that these objectives were met in the course.
Next steps: You love Python and want to learn more. (1)

1. Check out more advanced topics such as:
   - Built-in functions (we only used a fraction of them in this course).
   - Exception Handling
   - Regular expressions
   - Graphical User Interfaces (GUIs)
   - Class (Object-Oriented) Programming

2. Use Python programming in your classes to complete your assignments.

3. Find reasons to write more programs. It’s a great workout for your mind. Check out Project Euler (http://projecteuler.net/) if you like mathematical problems.

4. Start a Puzzle Solving club.

5. Check out Google Summer of Code.


7. If you have other ideas, please share them with me.
Next steps: You really enjoyed the problem-solving aspects of the course. (2)

1. Have you thought about being a computer science (CS) major or minor?
   - We would love to have you.
   - CS is all about computational problem solving. CS is not about just about programming.
   - However, programming is a major component in the computer scientist’s arsenal of tools.
   - If you want to know more about a CS major or minor, please feel free to talk to me about it.

2. When you find yourself with an interesting problem, write a program to see if it helps you solve it.
Next Steps: You hate Python and you never want to have anything to do with programming ever again. (3)

Well, that’s fine too.
That’s all folks!

I had fun! I hope you did too. 😊