CSCE 110 — Programming I
Final Remarks

Dr. Tiffani L. Williams

Department of Computer Science and Engineering
Texas A&M University

Spring 2014
Final exam information

- Final exam review: Thurs., May 1st at 9:30 in this classroom.
- Final exam: Friday, May 2nd from 12:30 to 2:30pm in this classroom.
- Final exam study material will be available on eCampus by the weekend.
- Complete the final survey about programming for 5 points added to your final exam score. The final survey is available in eCampus (read course announcements).
- Also, if you have not filled out the evaluation for this class 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, Rock-Paper-Scissors, Blackjack Dice, Craps, Shut the Box, Words in a Word
   - **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)
   - **Computer Art:** Random rectangles, visualizing binary strings
What a great journey we have taken! (2)

- You have written a lot of Python programs this semester.
- Sometimes you were asked to write programs that were straight-forward.
- Other times, you were asked to write programs that hopefully challenged your problem-solving skills.
How many Python programs will you have written in this course?

- Exams: 3

Exams: 3
How many Python programs will you have written in this course?

- Exams: 3
- Exam sample questions: 10

Total: You have written at least 100 programs!
How many Python programs will you have written in this course?

- Exams: 3
- Exam sample questions: 10
- Quizzes: 6

Total: You have written at least 100 programs!
How many Python programs will you have written in this course?

- Exams: 3
- Exam sample questions: 10
- Quizzes: 6
- Labs: 28

In-Class: 47
Other (e.g., false starts): >= 6
Total: You have written at least 100 programs!
How many Python programs will you have written in this course?

- Exams: 3
- Exam sample questions: 10
- Quizzes: 6
- Labs: 28
- In-Class: 47

Total: You have written at least 100 programs!
How many Python programs will you have written in this course?

- Exams: 3
- Exam sample questions: 10
- Quizzes: 6
- Labs: 28
- In-Class: 47
- Other (e.g., false starts): \( \geq 6 \)?

Total: You have written at least 100 programs!
How many Python programs will you have written in this course?

- Exams: 3
- Exam sample questions: 10
- Quizzes: 6
- Labs: 28
- In-Class: 47
- Other (e.g., false starts): $\geq 6$?
- Total: You have written at least 100 programs!
Each of you took ownership of your thoughts in order to solve a 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+ young minds think differently to solve the same problem!
Lab Difficulty vs. Time

Average Time Required/Difficulty for the Labs

Lab Number

Time Required (Hours)

Average Time Required/Difficulty for the Labs

Lab Number

Time Required

Difficulty

1

2

3

4

5

6

7

8

9

10

Tiffani L. Williams (Texas A&M)

CSCE 110

Spring 2014
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.
- Many of you gave a more satisfactory definition.
“Programming is analogous with a Rubik’s cube. There is one goal but many unique combinations of ways to accomplish it.

— CSCE 110 student, Spring 2014
“Programming is input plus instructions, and it gives you an output. But really, programming has opened my eyes to have attention for detail, and to think outside of the box. At times, it is very difficult, but I like it!

— CSCE 110 student, Spring 2014
“Programming is innovative, where I’m not afraid to try anything just to see what happens.

— CSCE 110 student, Spring 2014
Programming is a puzzle
Programming is a tool.
If you know how to do it
you’ll be really cool, that’s bool!

— CSCE 110 student, Spring 2014
“Programming is a way of getting a computer to show you how vague your instructions really are.

— CSCE 110 student, Spring 2014
def What_is_Programming():
    print 'Programming is being able to complete'
    print 'a task with a set language that allows'
    print 'you to find the solution.'

main()

— CSCE 110 student, Spring 2014
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. 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. 😊