Computer Programming for Non-Majors

Matthew X. Curinga

Table of Contents

**Computer Science 0145-160, Srping 2016**

**Keywords:** computer programming, CS1, python, computational thinking, critical computational literacy

**Description:** This course introduces students to programming and some core concepts of computer science, using a modern, object oriented programming language (currently Python). Students learn concepts of variables, functions, repetition/loops, basic data structures (arrays, lists, hashtables), and basic object oriented programming. They also consider the role of computation, software, and technology in our society.

**Class meetings:** Tues. & Thurs. 12:15-1:30PM, Room 100, Swirbul Library (Garden City)

**Instructor**

* [Matthew X. Curinga](http://matt.curinga.com), [mcuringa@adelphi.edu](mailto:mcuringa@adelphi.edu)
* [Post Annex, Room 1](http://goo.gl/maps/XReYB)

**Dr. Curinga’s Office Hours**

* Monday, 11-1:00PM
* Tuesday, 2:30-4:30PM
* Thursday, 3-5PM
* *office hours by appointment*

# Learning Goals

* understand the types of problems that can be solved using computational techniques
* understand the basic concepts of computation (CPU, RAM, permanent storage, GUIs, file systems, network connections)
* learn core computer programming concepts (abstraction, variables, conditions, functions, repetition, recursion)
* think algorithmically to design and test computer programs
* master the basic syntax and idioms of the Python programming language
* use technical documentation, APIs, and the internet to learn new technical concepts
* develop step-by-step problem solving and debugging practices

# Required Texts

[*Think Python: How to think like a computer scientist.*](http://openbookproject.net/thinkcs/python/english3e/) by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers. [Free online]

Poitras, L. (2014). *Citizenfour*. [Watch online](https://thoughtmaybe.com/citizenfour/) or HBO Go

Taylor, A. (2015). *People’s platform: taking back power and culture in the digital age.* New York: Picador, Henry Holt and Company. [Selections on Moodle]

# Bibliography

*This is a selected bibliography of computer science and Python texts and other materials that you may explore as references or further reading.*

Alvarado, C., Dodds, Z., Kuenning, G., & Libeskind-Hadas, R. (2013). [*CS for All*](http://www.cs.hmc.edu/csforall/). Claremont, CA: Harvey Mudd College.

Barry, P. (2009). *Head first programming: [a learner’s guide to programming using the Python language]*. Beijing ; Sebastopol, CA: O’Reilly.

Downey, A. (2012). *Think Python*. Sebastopol, CA: O’Reilly. [free](http://www.greenteapress.com/thinkpython/html/index.html) [py v3](http://faculty.stedwards.edu/mikek/python/thinkpython.pdf)

Pilgrim, M. (2009). [*Dive into Python 3*](http://www.diveinto.org/python3/index.html). New York: Apress.

[*The Python Tutorial v.3.3*](http://docs.python.org/3/tutorial/).

Zelle, J. (2010). *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates Inc.

# Technology Requirements

Everyone will need access to a Python development environment outside of class in order to complete the homework and project assignments. We will be using the following core tools that you can install for free:

* [Python 3](https://www.python.org/downloads/): the python language, you may have an earlier version installed (e.g. Python 2.7), but you will need Python 3 (Python 3.3, 3.4, etc) for this course
* [PyCharm Community Edition](https://www.jetbrains.com/pycharm/download/): PyCharm is an Python IDE which we will use to write and run our code.

# Class meetings

We will have 2 class meetings each week. You should complete the

| Week | Date (Tues) | Topic | Reading | Lab (Thurs) |
| --- | --- | --- | --- | --- |
| 1 | 1/26/16 | Variables & Statements | TIP 1 & 2 | Quiz 1 |
| 2 | 2/02/16 | Functions | TIP 3 & 4 | Quiz 2 |
| 3 | 2/09/16 | Conditionals | TIP 5 | Quiz 3 |
| 4 | 2/16/16 | More Function | TIP 6 | Quiz 4 |
| 5 | 2/23/16 | Iteration/Loops | TIP 7 | Quiz 5 |
| 6 | 3/01/16 | Strings | TIP 8 | Quiz 6 |
| 9 | 3/08/16 | *Citizenfour* | *Citizenfour* | (no quiz) |
| - | 3/15/16 | *No Class. Spring Break* | - | - |
| 7 | 3/22/16 | Review | - | Practice Exam |
| 8 | 3/29/16 | Midterm | - | **In Class Midterm** |
| 10 | 4/05/16 | Tuples & Lists | TIP 9 & 11 | Quiz 7 |
| 11 | 4/12/16 | Modules & Files | TIP 12 & 13 | Quiz 8 |
| 12 | 4/19/16† | Internet & Society | *People’s platform* | (no quiz) |
| 13 | 4/26/16 | Classes & Objects | TIP 15 & 16 | Quiz 9 |
| 14 | 5/03/16 | Recursion | TIP 18 | Quiz 10 |
| 15 | 5/10/16 | Exceptions & Dicts | TIP 19 & 20 | - |
| 16 | 5/17/16 | Final Review | - | Review |
| 17 | 5/24/16 | Final Exam | - | **Final Exam** |

*† No class on Tuesday in week 12.*

* TIP: *Thinking in Python*

# Assignments and Grading

## Grading structure

| Assignment | Pct |
| --- | --- |
| Participation | 10% |
| Quizzes | 20% |
| Midterm | 30% |
| Final Exam | 40% |

## Participation

Students are expected to attend every class session, to come prepared having done the reading and practice exercises, and to participate fully: working individually, in pairs, and in teams during in-class lab activities.

## Quizzes

There will be 10 quizzes which will be completed individually at the start of class. Quizzes consist of 2 short answer programming questions drawn directly from the reading for the week. Quizzes are designed to be passed easily by students who are keeping up with the reading and the concepts in the course, and will help students and the instructor maintain a good pace for the development of the course. All quizzes will be open book/open computer. Students can use the internet to read documentation and can use PyCharm, IDLE, and other tools to test their code. No communication with other students is allowed, and no direct help from outsiders will be allowed during quizzes. Quizzes will be timed and work will be submitted “as is” when time expires.

## Midterm

There will be an in-class midterm for this course on Thursday of Week 8. The midterm will require most/all of the available class time. The exam will consist of a short answer section as well as a full program which will likely require several functions to solve. There will be no partial credit for short answers. The full program will be scored with partial credit. Students will choose from one of two possible problems for the long-answer. The exam will be “open book”—students can use their notes, online resources, the course text, code examples from class, etc. The only exception is that real-time communication between students, or between students and outside parties is prohibited during the exam (i.e. no Slack, IM, texting, IRC, posting new questions to forums, etc).

## Final Exam

The midterm will test students’ abilities to synthesize the various aspects of computer programming to solve problems creatively. There will be no short answer questions. Students will choose one of several challenges to answer. They will write one complete program that offers a solution to the problem. The problems will not have a single, “right” answer, rather they will require analysis and creative thinking on the part of the students. Their solution will be evaluated on the following criteria:

* **15 points, code:** is the code efficient and elegant? do students demonstrate effective use of Python functions, libraries, and data structures? is the code organized into functions? does it use function parameters for abstraction and to make code reusable?
* **10 points, solution:** how well does the program solve the problem? does it work in all cases? to what degree is it bug free?
* **10 points, creativity and risk:** does the program offer a creative solution to the problem? does the student demonstrate that they have studied programming outside of class? are they able to read documentation to incorporate new ideas into their work?
* **5 points, style:** is the code well formatted with white space? are functions and variables named clearly and concisely? are comments used where needed and omitted when not?