

CS 15 Spring, 2013
Fundamentals of Computer Science I

Professor: Dr. Gretchen Ostheimer

Office: Adams 207B

Phone: 516-463-6106

Email: Gretchen.Ostheimer@hofstra.edu

Web site: <http://www.cs.hofstra.edu/~cscgzo/>

Office hours: Tuesday, Thursday 12:45-2:15pm,
and by appointment Tuesday, Wednesday, Thursday

Textbook: *Python Programming: An Introduction to Computer Science*, Second edition, John Zelle. Franklin, Beedle and Associates.

1 Overview

This course provides an introduction to the fundamentals of computer science: problem specification, algorithm design and algorithm analysis. Students will learn basic programming concepts (variables, assignment, control statements and functions). Students will work with a programming language called Python, a language which has proven to be suitable for beginning students and which is also currently in use in industry: for example, Python is the language of choice for NASA's main shuttle support contractor!

Computer Science is a broad discipline that encompasses many diverse areas of research. In order to give students an experience of this breadth, in this class we will explore one area of research that is of particular interest right now: cryptography. Cryptography is the study of making and breaking codes. Cryptography is essential whenever data must be kept private: financial records, medical records, and military information are but a few examples. We will explore cryptography from a wide range of perspectives:

- What kinds of algorithms exist for encrypting and decrypting information, and what are their relative strengths and weaknesses?
- What kinds of mathematics arise in cryptographic algorithms?
- What are the political and social implications of strong cryptographic systems?
- What is the history of cryptography and how has this history affected the direction of computer science?

2 Learning Objectives

Liberal Arts Objectives:

1. to learn to solve unfamiliar problems (without being taught how to solve them!);
2. to think clearly and analytically;
3. to work cooperatively;
4. to read closely; to write and speak precisely.

Computer Science Objectives:

1. to understand what is meant by problem specification, algorithm description and algorithm analysis;
2. to understand the classification of problems into tractable, intractable and undecidable, and to be able to provide examples of each type;
3. to appreciate the practical importance of this complexity classification scheme;
4. to understand the difference between secure and insecure cryptography systems; and between public key and private key cryptography systems;
5. to understand why complexity analysis is important in the development of cryptography systems;
6. to be able to use technical language in writing and speaking precisely and accurately about problems in computer science.

Programming Objectives:

1. to be able to read a description of a problem, and
 - (a) develop an algorithm to solve it;
 - (b) use functions, parameters, decision structures and loops to implement an algorithm;
 - (c) develop a reasonable suite of examples on which to test a program;
 - (d) use a programming style that conforms to the standards developed in class.
2. to be able to write a Python program using
 - (a) assignment, input and print statements,
 - (b) conditional statements,

- (c) while and for loops,
- (d) lists and strings,
- (e) function calls, with parameters and return statements,
- (f) recursion,
- (g) nested loops and lists of lists,
- (h) objects.

3 Graded Work

Programming: You will be given a series of programming exercises designed to help you achieve the programming objectives above while reinforcing the computer science objectives. The extent to which you have done so will be determined in three programming assessments. The grade you will get will depend on the programming objectives which you can demonstrate that you have achieved by the final assessment.

To get a *C* for this portion of your grade, you must achieve programming objectives 1 and 2a through 2e. To get an *A*, you must achieve all of the objectives. Furthermore, you must demonstrate that you can take a complex problem, break it into subproblems, and design and implement suitable functions solving those subproblems. Partial progress toward the *A* level objectives will result in a grade between C^+ and B^+ .

Please note that this grading system is highly non-standard. For example, I don't expect *anyone* to achieve *A* level work until the very end of the semester, since the learning objectives are chosen in such a way that the entire semester will be needed to achieve them. Also note that the grade is based entirely on what you can demonstrate in the scheduled assessments. If at any stage during the semester you start to worry about your grade in the class, it is imperative that you come talk to me. We will look over your work together, and assess how you are doing vis-a-vis the learning objectives.

Lab Reports: Three times during the semester, you will write a lab report summarizing the progress made so far in the semester. Precise instructions will be handed out in class. At the end of the report, you will acknowledge all the help you got on the labs, whether from me, from a tutor, from a friend, from the textbook, from the web, or from any other source. Note that failure to properly acknowledge such help violates Hofstra's Academic Honesty policy.

The final grade for this part of the course will be based entirely on the final lab report, but if you failed to hand in at least one earlier report, that final grade will be docked by 10 out of 100 points. Thus, each student can fail to hand in one lab report with no adverse impact on the grade. Note that late lab reports are not accepted under any circumstances.

Group Work Reports: The computer science topics explored in group work are very important. You will be asked to write several reports summarizing your findings in group work – I will hand out precise instructions for each such report. The lowest grade will be dropped and the remaining grades will be averaged for this part of the grade. At the end of the report, you will acknowledge all the help you got in group work or in writing your report, whether from me, from a tutor, from a fellow student, from a friend, from the textbook, from the web, or from any other source. Note that failure to properly acknowledge such help violates Hofstra’s Academic Honesty policy.

Note that late group work reports are not accepted under any circumstances.

Quizzes: Regular quizzes will be used to test your understanding of all the concepts discussed in class. These include the computer science and programming topics, as explored in the readings, in group work and in lab. To calculate the grade for this portion of the class, I will drop the lowest quiz grade and average the rest. This way if you miss a class due to illness, your grade will not suffer. Note that I do not give make-up quizzes under any circumstances.

Exams: There will be two midterm exams. There will be a comprehensive exam during the scheduled final exam period. This exam will test the extent to which you have achieved all of the objectives of the course. If an emergency arises and you must miss the final exam, contact me immediately. If you have a valid reason for missing the final (which you can document appropriately), then you will be given an incomplete, and you will take the final at the start of the next semester.

Your exam grade is calculated by writing down the two midterm grades and then writing down your final exam grade twice. I will then drop the lowest of these four grades, and average the remaining three. Note that I do not give make-up exams under any circumstances.

Grade Breakdown:

1. lab reports **10%**
2. programming assessments **30%**
3. group work reports **10%**
4. quizzes **20%**
5. midterms and final exam **30%**

Attendance, Lateness and Makeup Policies Students are required to come to class, to arrive on time, to stay in the classroom for the duration of class, and to abstain from such disruptive behaviors as checking email or surfing the web. Cell phones must be turned off before entering the classroom.

Since it does happen that sometimes a student is too sick to attend class, each of you will be assigned a “buddy” . If you miss a class, it is your job to get the notes and the handouts from your buddy before the next class meeting. If your buddy misses a class, it is your job to take careful notes, and to pick up extra handouts for your buddy.

Academic Honesty: Students enrolled in this class implicitly promise to adhere to Hofstra’s policies regarding academic honesty. Whenever you consult anyone (another student in this class, another student not in this class, a tutor, an instructor, anyone) or any other source (the textbook for the class, another book, a web site, anything), you must acknowledge the help you received in writing. It is expected that *every* piece of work you hand in will have an acknowledgments section. Students who are found to have violated their promise (either by cheating or by assisting another student in cheating) are given a 0 on the given assignment, are reported to the dean, and may, furthermore, receive an automatic F in the class.

4 Important Dates

Withdrawal dates:

- Last day to withdraw from course without **W**: February 25
- Last day to withdraw from course with **W**: April 15

Lab reports are due on the **programming assessment** days:

- Wednesday, February 27
- Wednesday, March 20
- Wednesday, May 1

Exams will be given on the following days:

- Midterm 1: Tuesday, March 5
- Midterm 2: Tuesday, April 2
- Final: to be announced

Quizzes and Group Work Reports: dates announced in class