CAS CS 111: Introduction to Computer Science I

Boston University, Fall 2018

Syllabus

Description: The first course for computer science majors and anyone seeking a rigorous introduction. Develops computational problem-solving skills by programming in the Python language, and exposes students to variety of other topics from computer science and its applications.

Prerequisites: none

Instructors

A1/B1: David G. Sullivan, Ph.D. (dgs@cs.bu.edu, PSY 228D, 64 Cummington Mall)
C1: Vahid Azadeh-Ranjbar, Ph.D. (vranjbar@bu.edu, PSY 223, 64 Cummington Mall)
See the course website for the schedule of instructor, TF and CA office hours.

Teaching Fellows/Assistants (TFs/TAs)

Nathan Cordner (ncordner@bu.edu)
Laura Greige (lgreige@bu.edu)
Justin Ingwersen (justini@bu.edu)
Eli Saracino (esaracin@bu.edu)
Reuben Xin Hong Tan (rxtan@bu.edu)

Course Assistants (CAs)

We are fortunate to have a number of undergraduate course assistants (CAs) as members of the course staff. They will be working with you in the labs and holding office hours each week. See the course website for their names and contact info.

Meeting Times and Places

Lecture: section A1: MWF, 10:10-11:00 am, STO B50
section B1: MWF, 12:20-1:10 pm, CAS 522
section C1: TuTh, 11:00-12:15 pm, CGS 511
Lab: a weekly, one-hour session; see your schedule for the time and location.
Important: You must also be able to take the midterm exams, which will be held on two Wednesday evenings (10/17 and 11/14) from 6:30-7:30 p.m. We will ***not*** meet on any other Wednesday evenings.

Course Website: http://www.cs.bu.edu/courses/cs111

Requirements and Grading

1. Weekly problem sets and final project (40% of the final grade)
2. Exams: two midterm exams (20%) and a final exam (30%)
3. Preparation and participation (10%)

To pass the course, you must earn a passing grade for each of these three components.

Collaboration Policy

You are strongly encouraged to collaborate with one another in studying the lecture materials and preparing for quizzes and exams.

Problem sets will include two types of problems:

- *individual-only* problems that you must complete on your own
- *pair-optional* problems that you may complete alone or with a partner.

For both types of problems, you may discuss ideas and approaches with others (provided that you acknowledge this in your solution), but such discussions should be kept at a high level, and should not involve actual details of the code or of other types of answers. **You must complete the actual solutions on your own** (or, in the case of a pair-optional problem, with your partner if you choose to use one).

Rules for working with a partner on pair-optional problems:

- You may *not* work with more than one partner on a given assignment. (However, you are welcome to switch partners between assignments.)
- **You may *not* split up the work and complete it separately.**
- **You must work *together at the same computer* for every problem that you complete as a pair.** While you are working, the screen should be visible to both of you. One person should type, while the other person observes, critiques, and plans what to do next. You should switch roles periodically, and your solution should be a true collaborative effort.
- You must *both* submit the same solution to each problem that you did as a pair, and you must clearly indicate that you worked on the problem as a pair by putting your partner's name at the top of the file.

Academic Misconduct

We will assume that you understand BU’s Academic Conduct Code:
http://www.bu.edu/academics/policies/academic-conduct-code

Prohibited behaviors include:

- copying all or part of someone else's work, even if you subsequently modify it; this includes cases in which someone tells you what you should write for your solution
- viewing all or part of someone else's work (with the exception of work that you and your partner do together on a pair-optional problem)
- showing all or part of your work to another student (with the exception of work that you and your partner do together on a pair-optional problem)
- consulting solutions from past semesters, or those found online or in books
- posting your work where others can view it (e.g., online).

Incidents of academic misconduct will be reported to the Academic Conduct Committee (ACC). The ACC may suspend/expel students found guilty of misconduct. ***At a minimum, students who engage in misconduct will have their final grade reduced by one letter grade (e.g., from a B to a C).***

Other Policies

Laptops: Students taking CS courses are expected to have a laptop capable of running a currently supported version of Microsoft Windows, Mac OS X, or Linux. See this page for more info: https://www.bu.edu/cs/undergraduate/undergraduate-life/laptops

Late problem sets: Problem sets must be submitted by the date and time listed on the assignment (typically by 11:59 p.m.). There will be a 10% deduction for submissions up to 24 hours late. **We will not accept any homework that is more than 24 hours late.** Plan your time carefully, and don't wait until the last minute so you will have ample time to ask questions and obtain assistance from the course staff.

Pre-lecture preparation: To help you prepare for lecture, you will typically be required to complete an assigned reading and/or watch one or two short videos. You will also be required to perform some type of brief task (an online quiz or other exercise) to demonstrate that you have completed the necessary preparation. Your work on these tasks will not typically be graded for correctness, but it should demonstrate that you have adequately prepared for lecture. The pre-lecture tasks must be submitted by the specified date and time. **Late pre-lecture work will not be accepted**.

The *attendance/participation* portion of your grade will be based on your consistent attendance at the lectures and lab sessions, and on your participation in the activities for each class. In particular, you must participate in small-group activities during lecture in which you will discuss questions with other students and "vote" on the answers. These activities are designed to deepen your understanding of the material, and you will be graded on your participation, *not* on the correctness of your answers. To accommodate unavoidable absences or lateness, you will receive full credit for attendance as long as you make at least 85% of the votes over the course of the semester and attend 85% of the lab sessions. **Voting from outside the classroom or voting for someone else is not allowed, and will result in a lowering of the participation grade of the students involved.**

The final exam will replace your lowest problem-set grade if doing so helps your final grade. (The final-project grade *cannot* be replaced.) The final exam will also replace your lowest midterm-exam grade if doing so helps your final grade.

The final grades are *not* curved. The performance of the class as a whole is taken into account in assigning letter grades, but this can only improve your grade, not harm it.

Extensions and makeup exams will only be given in *documented* cases of serious illness or other emergencies.

You cannot redo or complete extra work to improve your grade.

Incompletes will not be given except in extraordinary circumstances.

Course Materials

Textbook: We will use the free textbook *CS for All* by Alvarado, Dodds, Kuenning, and Libeskind-Hadas. It can be found here: http://www.cs.hmc.edu/csforall/index.html

In-class software: We will be using the Top Hat platform for in-class activities and attendance. More detail will be provided in class.

Schedule (tentative)

week	lecture dates	topics, exams, assignments, and special dates
0	A/B: 9/5, 9/7 C: 9/4, 9/6	Course overview and introduction Computing fundamentals with Scratch *No labs this week.*
1	A/B: 9/10, 9/12, 9/14 C: 9/11, 9/13	Getting started in Python Data types and expressions Strings and lists A first look at functions Making decisions (conditional execution) **Problem Set 0 (all) due on 9/16**
2	A/B: 9/17, 9/19, 9/21 C: 9/18, 9/20	Functions (cont.) Local and global variables; the runtime stack Recursion *9/17: last day to add a class* **Problem Set 1, part I due on 9/20** **Problem Set 1, part II due on 9/23**
3	A/B: 9/24, 9/26, 9/28 C: 9/25, 9/27	Higher-order functions and list comprehensions Recursive design **Problem Set 2, part I due on 9/27** **Problem Set 2, part II due on 9/30**
4	A/B: 10/1, 10/3, 10/5 C: 10/2, 10/4	Lists of lists; encryption and decryption Algorithm design Representing information **Problem Set 3, part I due on 10/4** **Problem Set 3, part II due on 10/7**
5	A/B: **10/9**, 10/10, 10/12 C: 10/11	Digital logic and circuit design *A/B: No lecture on 10/8 (Columbus Day)* *A/B: Lecture on 10/9 (Mon. schedule)* *C: No lecture on 10/9 (Mon. schedule)* *No labs this week.* *10/9: last day to drop without a 'W'* **Problem Set 4, part I due on 10/11** **Problem Set 4, part II due on 10/14**
6	A/B: 10/15, 10/17, 10/19 C: 10/16, 10/18	Digital logic and circuit design (cont.) Assembly language **Midterm 1 on 10/17 from 6:30-7:30 pm** **Problem Set 5, part I due on 10/21**
7	A/B: 10/22, 10/24, 10/26 C: 10/23, 10/25	Assembly language (cont.) Loops and imperative programming Cumulative computations **Problem Set 5, part II due on 10/28**

8	A/B: 10/29, 10/31, 11/2 C: 10/30, 11/1	Design using loops Nested loops References; mutable vs. immutable data **Problem Set 6, part I due on 11/1** **Problem Set 6, part II due on 11/4**
9	A/B: 11/5, 11/7, 11/9 C: 11/6, 11/8	2-D lists Object-oriented programming File processing Classes: creating your own types of objects *11/9: last day to drop a class with a 'W'* **Problem Set 7, part I due on 11/8** **Problem Set 7, part II due on 11/11**
10	A/B: 11/12, 11/14, 11/16 C: 11/13, 11/15	Classes: creating your own types of objects (cont.) Dictionaries **Midterm 2 on 11/14 from 6:30-7:30 pm** **Problem Set 8 (all) due on 11/18**
11	A/B: 11/19 C: 11/20	Games and AI *A/B: No lectures on 11/21, 11/23 (Fall break)* *C: No lecture on 11/22 (Fall break)* *No labs this week.*
12	A/B: 11/26, 11/28, 11/30 C: 11/27, 11/29	Games and AI (cont.) Inheritance Overview of the final project Finite-state machines **Problem Set 9, part I due on 11/29** **Problem Set 9, part II due on 12/2**
13	A/B: 12/3, 12/5, 12/7 C: 12/4, 12/6	Finite state machines (cont.) Algorithm efficiency and problem "hardness" **Problem Set 10 (all) due on 12/6**
14	A/B: 12/10, 12/12 C: 12/11	Problem "hardness" (cont.) Wrap-up **Final project due on 12/11** *12/13-16: Study period*
15		**Final exam: date and time TBD** **Please wait until your instructor informs you of the date. The initial date posted by the Registrar will not be correct.** ***Make sure that you are available for the entire exam period – up to and including Friday evening, December 21!***