

CS 101 Computer Science I

Fall 2013

Instructor Muller

Syllabus

Welcome to CS101. This course is an introduction to the art and science of computer programming and to some of the fundamental concepts of computer science. Students will write programs in the Python programming language. Good program design methodology will be stressed throughout. There will also be a study of some of the basic notions of computer science, including computer systems organization, files, and some algorithms of fundamental importance.

Course Goals

The main goal of this course is to help the student develop an understanding of *computation* and to help them master the art of designing algorithms, and developing the programs that implement them. Important parts of the latter include documenting and testing the program.

Students will learn how to decompose problems into specific subproblems, write an algorithm to solve a specific problem, and then translate that algorithm into a Python program.

Basic Information

CS101 has two 75-minute lectures each week and one one-hour lab. Lectures for section 3 are in Fulton Hall 250 at 12PM. Lectures for section 4 are in Fulton Hall 250 at 1:30PM. Attendance at the lectures is critical, as all new material will be presented there. **Note:** Laptops are not allowed in lecture. If you have special circumstances that require you to bring one, please see me.

Course Web Site

Please bookmark the course homepage:

<http://www.cs.bc.edu/~muller/teaching/cs101/f13/>

We will use this site very heavily throughout the semester and most of the course materials will be distributed through this site. Some lab materials will be distributed through the linked Piazza site. Problem sets are to be submitted through the linked BB/Vista website. Note that we will not have a course homepage on BBVista.

Lab Times

All lab/section meetings are in Maloney Hall 550. The labs are run by the course Teaching Assistants. They provide a great opportunity for you to work with a partner and practice with new material on some fun problems in a supervised setting. Regular attendance at labs is strongly encouraged. **Please bring your laptop if you have one.** If you do not have a laptop, please see me.

Lab times are:

1. Section 00802, (Danny) Meets Tuesdays 4PM,
2. Section 00702, (Milo) Meets Tuesdays 5PM,
3. Section 00801, (Meg) Meets Wednesdays 3PM,
4. Section 00701, (Jon) Meets Wednesdays 4PM.

Staff

Instructor: Robert Muller, robert.muller2@gmail.com, office: Maloney Hall 570, hours: Mondays 10AM - 12PM, Thursdays 3PM - 5PM, 617-552-3964.

Head Teaching Assistant: Meg Bednarcik, meg.bednarcik@gmail.com, office: Fulton 160, hours: TBA.

Teaching Assistant: Jon Hegarty, jon.hegarty@bc.edu, office: Fulton 160, hours: TBA.

Teaching Assistant: Danny Schlitt, schlitda@bc.edu, office: Fulton 160, hours: TBA.

Teaching Assistant: Milo Watanabe, milo.watanabe@bc.edu, office: Fulton 160, hours: TBA.

Problem Sets

Each week you will be assigned a problem set. Unless otherwise specified, all problem sets are due on 12PM on the specified due date. The single best indicator of success for computer science is *starting problem sets early*.

Problem sets should be submitted for grading by uploading an appropriately named zip file through the course Blackboard/Vista web site. (As linked from the course home page.) Problem sets cannot be submitted as email attachments. Attempts to submit problem sets as email attachments will not receive an email reply indicating that the attempted submission failed.

Topics

Roughly construed and subject to variation.

1. Introduction and admin. The Python shell. Basic types `int` and `float`. Constants, variables and expressions. Evaluation, values. Function definitions and calls. Libraries, the `stdraw` library.
2. The *replacement model* of evaluation. Using variables to name values. The `bool` type, comparing, branching. Conditional expressions and statements. Repetition (using conditionals and recursive function calls). Divide and Conquer. Tiling the unit square.
3. More on repetition, primality. Approximation algorithms, `sqrt`, `pi`. Nested definitions, block-structure and scope.
4. Structured data types. Lists and tuples. Indexing, slices, concatenation, list comprehensions, tuple patterns, zipping and unzipping. Functions are values. Lambda, `map` and `reduce`.
5. **First Exam.** Comparing and Searching. Linear search, binary search, an introduction to algorithmic complexity. Binary search trees.
6. Sorting. Mergesort and quicksort. Dictionaries. Association list implementation of dictionaries, binary search trees for dictionaries.
7. *Storage*, persistent and volatile, system and user, static and dynamic, the stack and the heap. Pointers and storage diagrams. The *storage model* of evaluation.
8. Bits and bytes, hexadecimal notation, data and program representation, the von Neumann machine, virtual machines.
9. **Second Exam.** Working with strings. Matching, coding, cryptography.
10. Application: digital audio. How hearing works, pulse-code modulation.
11. Application: digital images. How sight works, color. Working with 2D arrays.
12. Mutation. Mutation-based repetition idioms `while`-loops and `for`-loops. Quicksort revisited.
13. Mutable dictionaries, working with strings and files, GUIs.
14. Defining new types, Abstract Data Types, APIs. Review and looking ahead.

Exams

There will be two midterm exams and a final exam. The exams are in class and are closed notes and closed book. You will have 75 minutes to complete each midterm, and 2.5 hours to complete the final. If you require extra time for documented reasons, please let us know.

Midterms

First Midterm Tuesday October 1,

Second Midterm End of October, date TBA,

Final Exam TBA.

Reading

We will be using a free on-line book CS for All, by Christine Alvarado (UC San Diego), Zachary Dodds (Harvey Mudd), Geoff Kuenning (Harvey Mudd), Ran Libeskind-Hadas (Harvey Mudd). This is the new edition of the book used for the intro course at Harvey Mudd. That course is similar to our course but not an exact match. See the course website for other references.

Grading

Your grade for this class will be a combination of your homework, exam, and participation work. Participation is largely based on effort (not correctness). Lab work will be incorporated into the participation score. Final grades are computed, roughly as follows:

- Ten problem sets, these account for 45% of your grade,
- Two midterm exams, each accounts for 12% of your grade and a final exam which accounts for 16% of your grade,
- Class, lab and piazza forum participation, together, these account for the remaining 15% of your grade.

Important! If you fail one of these components, you will fail CS 101, even if your weighted-average scores are mathematically above the passing threshold.

Late Homework Policy

Homework is due on the day indicated at 12PM. *This is a strict deadline.* Homework submitted at 12:01PM is one day late as is homework submitted 23:59 late. Late homework is penalized 25% per 24-hour period.

In the case of medical exigencies, students may petition the Head TA or the Instructor for an extension. Medical problems or family emergencies are the only conditions under which extensions will be granted.

Honor Code

All solutions and code should be produced by you alone, or by you and a partner, where appropriate. For pair-programmed assignments, each partner needs to submit the assignment and each needs to acknowledge the other partner when submitting.

You may discuss algorithms at a high level with any student in the class. You may also help any student find a small bug in their code. However, you may not copy solutions from anyone, nor should you collaborate beyond high-level discussions with anyone who is not your partner. For pair programming problems, you must follow the guidelines given above.

If you have any questions about what behavior is acceptable, it is your responsibility to come see one of the instructors before you engage in this behavior. We are more than happy to answer any questions you may have.