CSCI 1100 Gateway to Computer Science

Fall 2022

Instructor Muller

Syllabus

Welcome to CSCI 1100. Gateway to Computer Science provides an introduction to the field of computer science for students with no coding experience and who may benefit from a review of the kinds of mathematical topics that tend to crop up in computing applications. An introduction to coding is a central theme of the course. Students will learn how to write applications that implement simple animations and games, digital audio and digital images. Gateway to Computer Science is an excellent preparatory course for the computer science introductory sequence CSCI 1101 Computer Science 1 and CSCI 1102 Computer Science 2.

Students will learn to code in the Python programming language. Good program design methodology will be stressed throughout the course. Students will also learn the unix command shell and how to work with the git system for sharing code and related digital artifacts. There will also be a study of some of the basic notions of computer science, including mathematical foundations and computer systems organization.

Course Goals

The main goal of the course is to lower the barrier to entry to the field of computer science and provide students with an understanding of *computation* and to help them master the art of designing algorithms, and developing the programs that implement them. 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

CSCI 1100 meets Monday through Thursday 2PM - 2:50PM in Stokes North Rm. 215. Attendance at both the class meetings is critical, as all new material will be presented there.

Course Web Site

The homepage for CSCI 1100 is on GitHub https://github.com/BC-CSCI1100/f22. (Students should certainly bookmark this link.) We'll use this site heavily

throughout the semester and most of the course materials will be distributed through this site. Problem sets are to be submitted through this site. Note that we will not have a course homepage on Canvas though we will use Canvas for posting grades.

Staff

Instructor: Robert Muller

email:robert.muller2@gmail.com

office: 245 Beacon St. Rm 508. Visiting? Happy to see you, please wear a mask.

hours: Tuesday 3PM - 5PM, Wednesday 4PM - 5PM.

phone: 617-552-3964

Teaching Assistant: Calista Agmata

email: agmata AT bc DOT edu

office: 245 Beacon St. Rm 122.

hours: Friday 5PM - 6PM, Saturday 11AM - 12PM.

Teaching Assistant: Andy Zheng

email: zhengno AT bc DOT edu

office: 245 Beacon St. Rm 122.

hours: Tuesday and Thursday 12PM - 1PM.

Problem Sets

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

Problem sets should be submitted for grading by using a `git push` command to upload them to the course GitHub site. 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. Overview and System Setup
2. Working with Unix the Command Line Interface
3. Python: Basic types & Expressions, Libraries & Functions
4. Working with Graphics, Repetition.
5. Repetition & Lists
6. Animation; Model-View-Update
7. More Animation
8. Numeral Systems & Digital Representations
9. Machines
10. Machines
11. Storage & Storage Diagrams
12. Working with Arrays
13. Working with 2-dimensional Arrays
14. Applications of Strings, Text & Files
15. Catching Up

Exams

There will be two midterm exams and a final exam.

Reading

There is one textbook for the course *OCaml from the Very Beginning* by John Whittington. There are two other books listed on the course homepage. We will use extensive code and lecture notes which will be posted to the course web site.

Grading

Your grade for this class will be computed on a 200 point scale from a combination of your problem sets, your exams, and participation work. Participation is largely based on effort (not correctness). Lab work and Piazza involvement will be incorporated into the participation score. Final grades are computed, as follows:

- Problem sets, these will account for 110 points;
- Three exams, 60 points;
- Class and piazza forum participation, together, these account for the remaining 30 points.

Late Homework Policy

Homework is due on the day indicated at midnight. *This is a strict deadline.* Homework submitted at 12:01AM 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 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.