Brandeis UNIVERSITY

Computer Science 10A
Introduction to Problem Solving with Python
Summer 2024

Contact Details

Instructor: Tim Hickey
Email: tjhickey@brandeis.edu

Meeting Times

Classes
Monday, Tuesday, Wednesday & Thursday 9:00 - 11:20 am
on Zoom

Office Hours
Office Hours for the professor and TAs will be posted on the course LATTE page

Course Description

Learning Goals:
This course aims to impart mastery over a set of fundamental programming concepts (or skills). The course will also teach students the fundamentals of the Python Programming language.

Teaching/learning strategies
Being a successful programmer is about more than just producing code that runs. Good programmers will have a solid foundational knowledge of the logic of programming that allows them to not only write code but to write *good* code. Additionally, programmers need to know how to problem solve and how to read and understand code written by others. This course is designed to support students learning from the ground up, starting with the fundamental concepts and building up to more complex applications. Students will be required to demonstrate foundational level understanding of concepts to pass the class, with higher grade outcomes linked to performance on assessments of higher level cognitive operations.

The material covered in this class has been broken down into a set of named skills. Most assignments in this class are graded pass/fail in terms of whether or not the student demonstrated mastery over a particular skill. Performance is tied to the number of skills that a student masters in three different categories.

Prerequisites
This is an intro course with no prerequisites, no major requirements. Because programming is such a fundamental skill, like reading and writing are for Literature, this course does not count towards the CS major or minor.

Credit Hours:
Success in this four-credit course is based on the expectation that students will spend about 24 hours of study time per week in preparation for class (reading, programming, practice, lectures, recitations)

Course Requirements

Academic integrity
Every member of the University community is expected to maintain the highest standards of academic integrity. A student shall not submit work that is falsified or is not the result of the student's own effort. Infringement of academic honesty by a student subjects that student to serious penalties, which may include failure on the assignment, failure in the course, suspension from the University or other sanctions (see section 20 of Rights & Responsibilities). Do not share your code with others in the class or you will be caught and sanctioned.

Comprehensive Weekly Skill Mastery Exams
We will release a take home exam at the end of every week, due the following Monday. The exams are made up of short answer questions that test basic knowledge of core programming concepts and skills. There will be one question for each skill covered up to that point. The questions are graded pass/fail for each question. Once you demonstrate you've mastered a skill by passing the corresponding question, you don't have to answer the questions for that skill in any future exams. If you don't pass a skill question one week then you'll have another chance to demonstrate mastery on that skill in each of the remaining weeks of the semester.

Weekly Programming Assignments
Programming Assignments will be assign during the week and due the following Monday. In addition to submitting code, you will be asked to submit 2 short (60 second) videos for each programming assignment - the first showing you running your program and and explaining why the results are correct and the second where you show your code and explain in plain English how the program works and why it is correct. Learning to talk about code and programming strategies is an important part of this course. These are graded pass/fail and if you don't pass then you can revise and resubmit as a regrade request.

Daily Participation Activities
A small part of your grade will be based on the number of participation activities you complete. These consist of in-class questions you answer with the Mastery Learning App and the exercises you complete in the Python3 Zybook during the assigned reading. These combine to make 1 skill.

Brandeis UNIVERSITY

Evaluation & Essential Resources

- Your course letter grade will be determined by the number of skills you master on the take home exams and the programming assignments and the participation exercises. The skills are listed on the Mastery Learning App front pages for Quizzes and for Programming Assignments, except for the participation skill. You demonstrate engagement by answering participation questions with a good faith effort and by making a good faith effort on the programming exercises in the Zybook we use as our textbook. These participation exercises together count as one skill.

Accommodations
Brandeis seeks to welcome and include all students. If you are a student who needs accommodations as outlined in an accommodations letter, I want to support you. **Please provide me with your documentation within the first 2 weeks of class.**

Course Materials
This course is partially based on the online textbook we will be using from Zybooks.
The link to buy/rent access to this book will be given on the Latte page.

If you are having difficulty purchasing course materials, please make an appointment with your Student Financial Services or Academic Services advisor to discuss possible funding options and/or textbook alternatives.

LATTE
LATTE is the Brandeis learning management system: http://latte.brandeis.edu. Login using your UNET ID and password. All course materials, assignments, lecture notes, videos, calendar, etc. will be linked through Latte.

This class also requires use of the Mastery Learning App (MLA) which will be linked on the LATTE page and requires you to sign up through your Brandeis email account.

TOPICS

The course will cover the following topics:

- Variables, Expressions, and built-in functions
- Boolean Expressions and Conditional Testing
- User-defined Functions, Parameters, and Return Values
- While Loops and Reasoning about Programs
- Lists, Indexing, Slices, and for loops
- Files and I/O
- Sets and Dictionaries
- List, Tuple, Set, and Dictionary Comprehensions
- Searching and Sorting
- Classes and Objects
- Python programs with Data Types
- Exceptions
- Effective Time Management
- Proper code documentation
- Clear explanations of problem solutions
- Use of VSCode IDE with Github Copilot
- Proper use of programming style
- Ability to solve problems with multi-step approach
- Use of Jupyter notebooks
- Ability to search for, install, and use packages
- Ability to describe and analyze algorithms using pseudocode