Programming Abstrations

Tyler Horan
horat351@newschool.edu
Computer Science
Tuesdays/Thursdays 13:00-14:00

Course Description

This course is the natural successor to Programming Methodology and covers such advanced programming topics as recursion, algorithmic analysis, and data abstraction using the Python programming language. Programming Abstractions assumes that you already have familiarity with good programming style and software engineering issues (at the level of Programming Methodology), and that you can use this understanding as a foundation on which to tackle new topics in programming and data abstraction. No textbook is required for this course. All materials can be accessed from this syllabus.

Topics

Abstraction and its relation to programming. Software engineering principles of data abstraction and modularity. Object-oriented programming, fundamental data structures (such as stacks, queues, sets) and data-directed design. Recursion and recursive data structures (linked lists, trees, graphs). Introduction to time and space complexity analysis. Uses the programming language Python covering its basic facilities.

Prerequisites

Solid performance in Programming Methodology and readiness to move on to advanced programming topics.

Course Materials

All course materials are available via the course website, which is accessible through your university portal. The assignments, handouts and lecture slides are available corresponding to each lecture on the syllabus.

Requirements & Expectations

You are required to attend and participate in class. There will be daily in-class quizzes, take-home assignments for discussion sections, and a final exam.

1. Lecture Attendance and Participation. We will start each lecture with a short quiz that you will take on your phone. You must be physically present in the room to take the quiz. Taking the quiz remotely is a form of academic dishonesty. For each question, you get half credit for answering incorrectly and full credit for answering correctly. You will also receive points for participating in classroom activities. After dropping the worst 10 percent of your responses, these points will account for 50 percent of your grade.

2. Discussion Section. You will do either readings, short assignments or both for discussion section each week. Any assignments must be turned in on time at section. Things happen so one missed assignment will not count against you. However, no late assignments will be accepted. Section will count for 30 percent of your grade in total, divided between attendance (10 percent), participation (10 percent), and assignments (10 percent).

3. Final Exam. The final will be held on Thursday, December 14, from 7:00pm to 10:00pm. If you cannot make this date and time, please drop the class. No alternate times or dates will be given. The final will count for 20 percent of your grade.

Policies

There are some University and classroom policies that you need to keep in mind.

1. Phones may only be used for designated class activities. I reserve the right to penalize your grade for misusing your phone in class.

2. Be honest. It is important that all students, and especially Freshers, understand what constitutes plagiarism in a written assignment or cheating on an exam. It is your responsibility to be aware of the university's standards. Cheating or other violations of academic integrity will be dealt with according to University policy. You should bear in mind that, while the University may punish violations of the community standard as it sees fit, final grades for classes are assigned by the faculty member teaching the course, not the University. I do not tolerate cheating.

3. Inform me in advance of excused absences. By design, attendance and participation are the most vital components of this course. If you have a University excused reason to miss class, please follow the proper procedure for letting me (or your TA, or both if necessary) know about it. You are responsible for knowing and acting in accordance with University policy.

4. Sign up for office hours if you have a concern. Section is for weekly discussion and review of the course material. If you have other concerns, I hold office hours on Mondays from 1pm to 2:30pm when classes are in session.

Schedule

Lecture 1: About CS & programming abstractions, the course philosophy, logistics of the course,introducing python.

Lecture 2: Data Types: Text, numeric, sequence, mapping, set, boolean, datetime and binary types.

Lecture 3: Data assignment and modification: Memory allocations, storage and access.

Lecture 4: Variables: Scope, access and un-assignment/reassignment.

Lecture 5: Functions: Definition, calling, inputs, outputs and libraries.

Lecture 6: Streams: Time, sequence and IO.

Lecture 7: Collections: Sets, Tuples and Iterations.

Lecture 8: Conditionals, operators and control flow: If, else, case and switch.

Lecture 9: Loops: For, map, while, do, break, and continue.

Lecture 10: Classes and Objects: Design, structure, and behavior.

Lecture 11: Inheritance: Hierarchies, parents and children.

Lecture 12: Linked Lists: Linearity, linkage, traversal.

Lecture 13: Binary Trees, Binary Search Trees: Search, traversal and insertion.

Midterm Exam

Lecture 13: Hashing: Concepts, performance, tables and types

Lecture 14: Recursion (Backtracking): Decision, optimization and enumeration.

Lecture 15: Recursion (Big O): Complexity, time and space.

Lecture 16: Recursion (Big O): O(1), O(N), O(N^x), O($2N^x$)

Lecture 17: Graphs: Explanation, implementation strategies and representation

Lecture 18: Polymorphism: Abstraction, inheritance and multiplicity.

Lecture 19: Sorting Algorithms: Selection, insertion, heapsort, merge and quicksort.

Final Exam