Introduction to Computer and Data Science Spring 2024

Basic Information

Duration: 14 weeks (the final exam is in Week 15)

Time zone: China Standard Time

Instructor: Xianbin Gu (xianbin.gu@nyu.edu);

Reference

- Lecture and lab slides/notes (uploaded in Resources every week)
- Starting out with Python(i.e., the Gaddis text). This book serves as a basic Python syntax reference.
- Introduction to Computation and Programming Using Python(i.e., the MIT text). This is the book used in MIT's undergraduate ICS course, see the link here.
- 9 Algorithms that changed the future(i.e., the 9-alg text). This provides a high-level overview of several algorithms at the CS landscape's core.

Development Environment

I recommend (Anaconda + VSCode) as the development environments. To use Anaconda + VSCode, please install Anaconda and VSCode, then in VSCode, please add the Microsoft Python extension and change the Python **interpreter** to the one that is labeled as ('base':conda). You may refer to this link: https://code.visualstudio.com/docs/languages/python.

Course structure

Seg1: Computer architecture with a history, Python basics, Quiz 1

Seg2: Algorithms and complexity families, Python OOP, Midterm

Seg3: Introduction to Data Science, Chat system, Quiz 2

Seg4: Advanced topics, Final

Course outline

Week	Topic	Lecture	Lab	Reference
Week 1	Computer Architecture	Foundations of Computer	Python review: syntax basic	Gaddis: Chapter 1(Arch. Intro)
Week 2	Computer Architecture	Computer Memory	Python review: mutable/immutable; built-in data structures	
Week 3	Computer Architecture	The Architecture	Python review: module, file operations	
Week 4	Algorithms	Algorithm basics, Sorting	Quiz 1	Guttag [MIT text]: Chapter 9, 10 ,11, 17, 18 MacCormack [9-alg text]: Chapter 1, 10 Python OOP notes

Week 5	Algorithms	Complexity analysis, Searching	OOP with Python	
Week 6	Algorithms	Algorithm design strategies	OOP with Python (with more examples)	Terminal commands and GitHub (video)
Week 7	Algorithms/ Midterm	Algorithm design workshop	The midterm takes place in the lab session. (no lab this week)	
Week 8	Data Science	Machine learning foundations, unsupervised learning	UP 1	Guttag [MIT text]: Chapter 12, 16, 19 (Machine Learning)
Week 9	Data Science	Supervised learning	UP 2	

Week 10	Data Science	Neural Networks and deep learning	UP 3	Goodfellow, Courville & Bengio [MIT] Deep Learning: Chapter 1, 2, 5 (ML Concepts), 6 (Feedforward Networks)
Week 11	Data Science	Reinforcement learning	Quiz 2, Final project intro	
Week 12	Advanced topics	Network, GUI Programming	Chat system with GUI	
Week 13	Advanced topics	Cybersecurity		MacCormack [9-alg text]: Chapter 4 (Public Key Cryptography)
Week 14	Final project	Final project office hours (tentatively)	Final project office hours	Submitting video records

Course objective

This course assumes relatively minimal prior CS background and programming experience. It is designed to hit three goals:

- The mastering of a modern object-oriented programming tool (Python), enough to bootstrap students and enable them to tackle real-world problems.
- The appreciation of computational thinking, a process that underlies the very foundations of any such solutions to the aforementioned problems.
- Providing an overview of the very diverse and exciting modern CS landscape.

To understand why there are these three goals instead of one, think for a moment like a cameraman. There is the camera (the programming tool), the mastermind behind it (computational thinking) and the world we want to take a shot at (the general CS landscape). We want you to understand and appreciate why Computer Science, a relatively young discipline of science by any measure, has so much potential for innovation and why you, too, should think of taking it seriously as a career.

How to study

- **Lectures/Recitations:** We will have classes weekly on Tuesday and Thursday (and Friday). Attendance at classes is mandatory. I may check class attendance sometime during the semester; missing one class results in a -0.5 points reduction in your final scores.

- **Office hours:** The time of office hours is flexible. Apart from the regular time slots, you can make appointments with the instructor.

- **Assignments:** You are expected to complete several assignments. All of them are Python programming problems and will be distributed on Gradescope.

- **Quizzes and Exams:** There are two quizzes and two exams. All will be distributed on Gradescope.

- **Unit Project:** You are expected to build a chat system individually. It will be built incrementally, with two building

blocks (group membership and indexing), and finally integrated into a complete whole.

- Final Project: You are expected to work in pairs towards developing an app based on the chat system you made in the unit project. The chat system acts as a communication substrate, on top of which, you are to implement additional functionality. For example,
  - Adding rudimentary encryption to protect against eavesdropping
  - Using the chat system as a backbone for the implementation of a simple multiplayer game
  - Making a graphic user interface for the chat system
- Programming environment: I recommend VSCode with Anaconda as the development tool. VSCode is a free source-code editor made by Microsoft and has become one of the most popular development tools since 2019, while Anaconda provides all packages we need in this course and can be maintained easily. However, other IDEs such as IDLE, PyCharm, and Anaconda Spyder are also fine for this course.

Grading Policy

Usually, the codes you submit (in assignments, unit projects, quizzes, and exams) will be graded with three levels:

- satisfactory: you get full points (i.e., your codes run smoothly, and the outputs are the same to expected)
- under-perform: you get half of the points (e.g., your codes run without any error raised, but not all of the outputs are the same to expected)
- unsolved: you get no point (your code cannot run; it raises errors in the runtime)

The portions of each item in the final score are listed below:

Item	Percentage in the final grade	Remarks

Assignments	5%	up to five compulsory assignments
Unit project	15%	complete a chat system in three UPs
Quizzes	10%	two quizzes 5% each
Midterm	20%	
Final	30%	
Final project	20%	extend the chat system

Here are the cut-offs for the final letter grades in previous semesters:

- A: >= 93
- A- : >= 90
- B+ : >=85
- B : >= 80
- C+ : >=75
- C : >=70
- D: >=60
- F : <60

This semester, the cut-offs should be very close to the above.

Midterm Evaluation

The midterm evaluation is based on the following Items:

- Assignments: 5%
- Quiz 1: 5%

- Midterm: 25%

The grade of the midterm evaluation is calculated as follows,

1. sum the scores of the above items with the weight respectively,
2. divide the sum by 0.35
3. convert the division to letters according to the cut-offs listed above.

Additional policies

Late policy and make-up policy: All assignments should be submitted before the due date. In general, I do not accept late assignments and do not offer substitute times for exams. Consideration may be given in case of special circumstances, such as a medical condition or family emergency, in accordance with university policy. If that case applies to you and you need additional time to complete an assignment or need a rescheduled exam, you should provide written documentation of your circumstances.

Communication: Email is occasionally used to make class announcements. It is your responsibility to check your NYU email account regularly. Assignments will typically be posted on Gradescope and NYU Brightspace. It is your responsibility to check Gradescope (and NYU Brightspace) for assignments and to submit your work in a timely manner.

Academic integrity: This class is bound by the Student Code of Conduct. The work you submit for grading should be a result of your own effort. This applies to all your work for this class, including assignments, projects, quizzes, and exams. It is normal to have problems in programming, and the best way to get help is to discuss them with the teaching staff. You can always turn to your professor and recitation instructor with any questions. However, **the following activities are never acceptable** and always constitute a violation of academic integrity:

- **Copying code or answers from others:** The work you submit must be written entirely by you. You may not copy code, or read code from another person directly or indirectly.

- **Allowing others to use your code or answers:** At no point should anyone else have access to your work. You may not show your work to others, even if they promise not to use it in their own work. You may use Github (and similar sites) as version control

storage, but it is your responsibility to ensure that your repository is private. If your code is publicly visible at any time, it constitutes a violation of this policy.