Introduction to Computer and Data Science (ICDS) - Spring 2025

Course Information

- **Duration: 14** weeks (Final exam on Week 15)

	Section 1			**Section 2**		
	Lecture	Recitation		Lecture	Recitation	
Sec.	001	003	004	002	005	006
When	Tue 9:45 – 11:35	Thu 9:45 – 11:35	Fri 9:45 – 11:35	Wed 9:45 – 11:35	Thu 9:45 – 11:35	Fri 9:45 – 11:35
Where	N208	SB230	E203	S308	SB124	E204
Who	Zhaonan	Sven	Yiyao	Zhaonan	Yiyao	Jennifer

- Teaching Crew:
  - Instructor: Zhaonan Wang *(zhaonan.wang@nyu.edu)*
    - Office hours @S730, Tue 3-5 PM
  - Teaching Assistant:
    - **Head TA**: Sven Simikin *(ss17116@nyu.edu)*
      - Office hours @S744
        - Wed 10-11 AM & 2:30-3:30 PM; Thu 2-3 PM
        - Book via NYU Connect
    - Yiyao Yang (*yangyiyao@sjtu.edu.cn*)
      - Office hours @SB124 11:35 AM-
    - Jennifer Feng (*jf3780@nyu.edu*)
      - Office hours @N725, Mon & Wed 3-4 PM
  - Learning Assistant (per ARC policy):
    - Steven Zhang *(qz2208@nyu.edu)*
      - Tutoring hours:
    - Mohamed Hendy (*mmh10013@nyu.edu*)
      - Tutoring hours:
- Resources:
  - Learning Management System (LMS): NYU Brightspace

        - If you are new to Brightspace, start with this page. At the first time you log in, please change the timezone in Account Setting to GMT+8: Shanghai
        - Lecture and lab slides/notes will be uploaded every week
        - CSCI-SHU 101 Spring 2025
    - Course registration system: NYU Albert
        - **Please attend the section you registered!**
        - A quick guide for freshmen
    - New scheduling system: NYU Connect
        - To talk to academic advisors

References

- [**OPTIONAL**] You do not need to buy any textbooks.
- 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

It is recommended to use (Anaconda + VSCode) as the development environments. To use Anaconda + VSCode, please install Anaconda and VSCode first, 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 Schedule

Week	Topic	Lecture	Recitation / Lab	Reference
Week 1 Feb 3-7	Computer Architecture	Foundations of Computer; Boolean algebra;	Python review: syntax basic	Gaddis: Chapter 1 (Arch. Intro)
Week 2 Feb 10-14	Computer Architecture	Computer Memory; principle of locality	Python review: mutable/immutable ; built-in data structures	
Week 3 Feb 17-21	Computer Architecture	The Architecture; Cycle of execution	Python review: module, file operations	
Week 4 Feb 24-28	Algorithms	Algorithm basics; pseudo-code; Sorting;	Quiz 1 (Computer Architecture)	Guttag [MIT text]: Chapter 9, 10 ,11, 17, 18 MacCormack [9-alg text]: Chapter 1, 10
Week 5 Mar 3-7	Algorithms	Complexity analysis; Searching;	Quiz 1 solutions; Object-oriented programming (OOP)	Python OOP notes

Week 6 Mar 10-14	Algorithms	Algorithm design strategies: Brutal-force, Divide and Conquer, Dynamic Programming;	OOP with Python (with examples)	Terminal commands and GitHub (video)
Week 7 Mar 17-21	**Midterm**	Midterm exam in-class	Midterm solutions; OOP with more examples	
Week 8 Mar 24-28	Data Science	Machine learning foundations; K-NN algorithm	UP 1	Guttag [MIT text]: Chapter 12, 16, 19 (Machine Learning)
Mar 31 – Apr 4: Spring Break				
Week 9 Apr 7-11	Data Science	Supervised & Unsupervised learning: regression, K-means;	UP 2	

Week 10 Apr 14-18	Data Science	Neural Networks and deep learning; LeNet;	UP 3	Goodfellow, Courville & Bengio [MIT] Deep Learning: Chapter 1, 2, 5 (ML Concepts), 6 (Feedforward Networks)
Week 11 Apr 21-25	Advanced topics	Network, GUI programming	Quiz 2 (Data Science); Final project intro	
Week 12 Apr 28 - May 2	Advanced topics	Cybersecurity: Diffie-hellman key exchange; RSA; Block-chain basic	**No recitation, no make-ups** on Apr 13 or Apr 27 Enjoy Labor Day!	MacCormack [9-alg text]: Chapter 4 (Public Key Cryptography)
Week 13 May 5-9	Advanced topics	Reinforcement learning; Q-learning	Chat system with GUI; Final project working hours	
Week 14 May 12-16	Final project	Final project office hours	Final project working hours	Final video recording deadline

Final Exam Week: May 19-23 (time & room TBA)

Course Objectives

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

- Mastering a modern object-oriented programming tool (Python) is enough to bootstrap students and enable them to tackle real-world problems.
- The appreciation of computational thinking is 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 science discipline by any measure, has so much potential for innovation and why you, too, should consider taking it seriously as a career.

How to Study

- **Lectures/Recitations:** We will have classes weekly: lecture on Tuesday/Wednesday and recitation on Thursday/Friday. Attendance at classes is essential to your learning. I may check class attendance sometime during the semester: from the second time on, each missing will result in a 1 point reduction to your final grade.

- **Office & tutoring hours:**
  - Entire teaching crew: instructor, Teaching Assistants
  - Academic Resource Center (ARC) Learning Assistants

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

- **Quizzes and Exams:** There are two quizzes and two exams.

- **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. Anaconda provides all the packages we need in this course and can be maintained easily. However, other IDEs such as IDLE, PyCharm, and Anaconda Spyder are also acceptable 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 as expected)
- underperforming: you get half of the points (e.g., your codes run without any error raised, but not all of the outputs are the same as 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:

Activity	% of Final Grade	Remarks
Assignments	8%	four assignments 2% each
Quizzes	12%	two quizzes 6% each
Midterm exam	20%	

Final exam	30%	
Unit project	15%	complete a chat system in three UPs
Final project	15%	extend the chat system

Here are the cut-offs for the final letter grades in this semester:

- A: >= 93
- A- : >= 90
- B+ : >=86
- B : >= 82
- B- : >= 78
- C+ : >=73
- C : >=68
- D: >=60
- F : <60

Midterm Evaluation

The midterm review will be conducted for academic advisors to evaluate your performance and progress during the semester. It will be based on the following Items:

- Assignments: 8%
- Quiz 1: 6%
- 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 0.39
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.

Use of Generative AI Tools: As this is a course designed to train your fundamental skills of programming and algorithmic thinking, you are asked not to use any GenAI tool (including assistive AI tools) in any programming-related activities. GenAI use is permitted to assist with activities such as brainstorming, outlining or similar, with the understanding that it is supplementary and not a substitute for personal engagement. Using GenAI to compose any part of assignment submissions or copy-pasting content for any course task is prohibited and violates academic integrity. All suspected instances of academic integrity violations will be referred to the Academic Affairs office and may result in failure of assessment tasks or the course.

Updated Feb 2025