SYLLABUS

**CIS 2300. Programming and Computational Thinking**

*Paul H. Chook Department of Information Systems & Statistics*

Zicklin School of Business, Baruch College, CUNY

Spring 2021

## Course Information

**Instructor**: Anh Luong

**Email**: anh.luong@baruch.cuny.edu

- Please put **CIS 2300 CMWA** followed by the main topic in your email subject

**Class Meeting:** Monday, Wednesday 10:45 AM - 12 PM via Zoom

**Office Hours**: Mondays 4 - 5 pm & by appointment

## Course Description

Computational thinking is an important skill for all students regardless of their business discipline. As such, this course covers fundamental computational concepts required for formulating and solving problems in programming. The goal of this course is to equip students with the ability to use computational principles such as abstraction, conditional, iteration, recursion, and functional decomposition.

*Note: Students cannot get credit for both CIS 2300 and MTH 3300. CIS majors will be required to take an additional CIS course if receiving credit for MTH 3300 to satisfy 24 credit requirements for the CIS major.*

## Course Objectives

Upon successful completion of this course, students will be able to:

- Understand and utilize a computational approach to solving problems
- Decompose bigger problems in smaller chunks and put them back together to solve bigger problems
- Develop comprehensive programs that can achieve useful objectives

## Course Materials

- **Software**: Python 3 & Jupyter Lab or Jupiter Notebook
- **Books**:
  - "Starting Out with Python", by Tony Gaddis, 3rd or 4th Edition
  - "Think Python", by Allen Downey, 2nd edition, 2015, O'Reilly
- **CodeLab** (free): codelab.turingscraft.com << You can use this website to further practice coding in addition to the required assignments. The exercises on here will not be graded.
  - Register for a student account
  - Join this course's section with the access code: TCAB-29015-WETX-47
- **Additional Materials** (readings, videos, tutorials, etc.) will be posted on blackboard

## Class Format

- We meet via Zoom every Monday and Wednesday, for 1 hour and 15 minutes each time.
- The meetings will include some combination of: lectures, code demos, discussion, & in-class assignments.

## Evaluation

The instructor reserves the right to curve the course grade if deemed necessary.

| Component | Weight | Details |
|---|---|---|
| **Assignments** (Weekly. In-class & homework) | 10% | - This component consists of open-ended questions &/or individual, pair, and group programming exercises.<br>- Assigned in class and as homework.<br>- This component is designed to reinforce the knowledge that you will learn from the class materials.<br>- ***Grading*** will be based on completeness. Solutions for the programming problems will be posted the week after the assignments are due. |
| **Projects** (2 projects) | 20% | - There will be **2** coding projects that require you to design and write programs that are more comprehensive than the weekly assignments.<br>- Each counts as 10%<br>- ***Grading*** will be based on correctness (80%) and code clarity/ readability (20%) |
| **Exams** (Midterm & Final) | 70% | - Exams will consist of T/F, multiple choice, and programming questions that require you to write and/or debug code. Exams will cover material from all aspects of the class sessions (lectures, programming exercises, discussions, videos, extra handouts and so forth).<br>- Midterm and Final exam each counts as 35%<br>- ***Grading*** for the programming questions will be based on correctness (95%) and code clarity/readability (5%) |

## Academic Integrity

No cheating will be tolerated in this class. If caught cheating, both the person who copied and the person who provided the solution will receive a score of **0**, either on the assignment/exam/project or for the class, depending on the seriousness of the incident. I also may choose to report the incident to the Dean.

You can discuss the assignments (NOT exams) with your classmates, friends, and/or consult resources from the internet, etc. But any work submitted must be your own and you need to be able to explain any part of your code when asked by me.

Importantly, I highly recommend you to first try to solve all problems on your own before looking up external resources. It's obvious that you can easily and instantly look up solutions online but it will not help your understanding and your programming skills if you approach every problem by looking up code written by someone else without first attempting it yourself.

The department of CIS fully supports Baruch College's policy on Academic Honesty, which states, in part: "*Academic dishonesty is unacceptable and will not be tolerated. Cheating, forgery, plagiarism and collusion in dishonest acts undermine the college's educational mission and the students' personal and intellectual growth. Baruch students are expected to bear individual responsibility for their work and to uphold the ideal of academic integrity. Any student who attempts to compromise or devalue the academic process will be sanctioned.*"

Additional information about the policy can be found at: http://www.baruch.cuny.edu/academic/academic_honesty.html

| **TENTATIVE SCHEDULE**<br>*(Subject to change. Announcements will be posted on Blackboard if changes occur)* | |
|---|---|
| **DATE** | **TOPICS** |
| **Week 1**<br>Feb 1, 3 | Course Overview & Introduction to Python |
| **Week 2**<br>Feb 8, 10 | Basic Data Types; Variables, Expressions, Statements |
| **Week 3**<br>Feb 17, 22 | Conditionals; Recursion |
| **Week 4**<br>Feb 24, Mar 1 | Iterations |
| **Week 5**<br>Mar 3, 8 | Functions |
| **Week 6**<br>Mar 10, 15 | Strings |
| **Week 7**<br>Mar 17, 22 | Lists<br>MIDTERM & PROJECT 1 DUE MAR 22 |
| **Week 8**<br>Mar 24, Apr 5 | Lists; Dictionaries and Sets<br>*Note: Apr 4 is the last day to withdraw with W grade* |
| **Week 9**<br>Apr 7, 12 | Dictionaries and Sets; Tuples |
| **Week 10**<br>Apr 14, 19 | Files and Exceptions |
| **Week 11**<br>Apr 21, 26 | OOP: Classes, Objects, Functions, Methods |
| **Week 12**<br>Apr 28, May 3 | OOP: Classes, Objects, Functions, Methods |
| **Week 13**<br>May 5, 10 | Inheritance |
| **Week 14**<br>May 12, 17 | Numpy<br>PROJECT 2 DUE MAY 17 |
| **TBA** | FINAL EXAM |