CIT 5910 Introduction to Software Development

Fall 2023

Instructor

Brandon Krakowsky, Research and Education Director at The Wharton School and Lecturer in Computer and Information Science at Penn Engineering.

Course Description

This course is an introduction to fundamental concepts of programming and computer science for students who have little or no experience in these areas. Includes an introduction to programming using Python, where students are introduced to core programming concepts like data structures, conditionals, loops, variables, and functions. Also provides an introduction to basic data science techniques using Python.

The second half of this course is an introduction to object-oriented programming using Java, where students are introduced to polymorphism, inheritance, abstract classes, interfaces, and advanced data structures. Students will also learn how to read and write to files, connect to databases, and use regular expressions to parse text. This course includes substantial programming assignments in both Python and Java, and teaches techniques for test-driven development and debugging code.

Course Learning Objectives

- Write programs in Python using basic programming language constructs and core data science techniques to analyze data
- Write large programs in Java using object-oriented programming techniques and good code design to separate parts of a program
- Identify and write a set of comprehensive test cases
- Use different development environments, tools, and frameworks for modern software development

Course Prerequisites

There are no prerequisites for this class. No prior programming background is expected.

Course Textbook

No Required Textbook.

Optional Books

Optional Books: These books are for students who would like to supplement the course material with additional reading(s). They are entirely optional.

Python

— Python Crash Course (by Eric Matthes): https://nostarch.com/pythoncrashcourse2e (You should also be able to find it on the O'Reilly site through the UPenn library)

— Think Python (by Allen B. Downey): https://greenteapress.com/wp/think-python/

— Automate the Boring Stuff with Python (by Al Sweigart): https://automatetheboringstuff.com/

— Python in Easy Steps (by Mike McGrath): https://www.amazon.com/Python-easy-steps-Covers-3-7/dp/1840788127/

Java

— Head First Java (by Kathy Sierra):

https://www.amazon.com/Head-First-Java-Kathy-Sierra/dp/0596009208

— Java in Easy Steps (by Mike McGrath):

https://www.amazon.com/Java-easy-steps-Mike-McGrath-dp-1840788739/dp/1840788739/ref=dp_ob_title_bk

— Java Concepts, Early Objects (by Cay Hortsmann)

https://www.amazon.com/Java-Concepts-Cay-S-Horstmann/dp/1119056500

Grading & Assessment

You must attempt all graded assignments to pass the course. If you have any questions or concerns about grading or progress in the course, please reach out to your instructor.

This course will use a variety of assessments to determine whether learners understand and can apply the key concepts and skills that the course teaches. This includes:

Type		%	Description
Homework Assignments	Homework 1	7.85%	In this course, you will have 8 assignments. Each assignment will require you to submit code using an external tool called Codio. Most assignments (except where noted) will be graded using a combination of 2 components : 1) Auto-grading: This is an automated grading set up beforehand. It will run in the back-end of Codio. In most cases, there will be a pre-submission test phase, which will include a handful of tests that you can run your code against to check its basic functionality. After you finalize your code and submit it, there will be a few more tests that run on your code, to give the final grade of this component. 2) Manual Grading: The TAs will also be manually grading your code. They will grade based on rubrics and criteria provided by the Instructor. You will finally receive just one grade which is a combination of auto-grading and TA-grading (except where noted).
	Homework 2	7.85%	
	Homework 3	7.85%	
	Homework 4	7.85%	
	Homework 5	7.85%	
	Homework 6	7.85%	
	Homework 7	7.85%	
	Homework 8	10%	
	Midterm Exam (Python)	15%	This will be a “take-home” exam. You will have the entire period to work on the exam during the availability dates (refer to the assignment schedule). Unlike regular assignments, you are not allowed to resubmit the exam, so be sure to double-check your submission before marking it as complete. Also, we do not normally grant extensions for exams. The exam will auto-submit when the deadline is reached. During the exam time, the course staff will not answer content-related questions on Ed Discussion or during Office Hours.

Exam	Final Part 1 (Java)	10.5%	Part 1 will be a “take-home” exam. You will have the entire period to work on the exam during the availability dates (refer to the assignment schedule). Part 2 will be a timed exam which will be closed book, closed notes, while being proctored by Honorlock. For both parts: Unlike regular assignments, you are not allowed to resubmit the exam, so be sure to double-check your submission before marking it as complete. Also, we do not normally grant extensions for exams. The exam will auto-submit when the deadline is reached. During the exam time, the course staff will not answer content-related questions on Ed Discussion or during Office Hours.
	Final Part 2 (Java)	4.5%	
Quizzes		5%	There are 24 quizzes that will be taken within Canvas.

Please read the instructions for each assignment very carefully to make sure you know where to submit to receive credit!

Grading/Curving:

- Homework assignment grades are returned 5–6 days after the due date.
- Exam grades are returned 10 days after the due date.
- Turning in an assignment early does not mean your grades will be released early.
- Grades for late-submitted assignments will be returned with the next set of grading.

At the end of this course, all final grades will be curved. Here's how it works: We visualize all final grades and observe the natural separation of the numeric grades into groups. Then we assign a letter grade to each group based on the numeric range. For example, depending on how well the class does overall, an A+ could be a 96 – 100. An A could be a 91 - 95. An A- could be an 87 - 90. And so on and so forth. These are just examples. This is typically beneficial to most students, and in no cases will this policy bring a final score down.

Below is an example of the typical fixed scale used for assigning letter grades to numeric grades. It's a good place to start, but based on the final curve, and as noted above, the grading groups will likely change.

97 ≤ A+ ≤ 100
93 ≤ A < 97
90 ≤ A- < 93
87 ≤ B+ < 90
83 ≤ B < 87
80 ≤ B- < 83
77 ≤ C+ < 80
73 ≤ C < 77
70 ≤ C- < 73
67 ≤ D+ < 70
60 ≤ D < 67
F < 60

Late Policy

An assignment that is turned in late will receive a 10% grade reduction per day. For any extraordinarily extenuating circumstances, please fill out the extension request form located under the Home page in Canvas. These extension requests must be submitted at least 24 hours before the assignment deadline, otherwise, your extension will be denied.

If your request is granted, you will see updated deadlines reflected in Canvas and Codio. You will need to fill out the extension form for each assignment you wish to have an extension. Please note we only approve extensions that are extraordinarily extenuating circumstances.

Regrade Requests

Regrade requests are handled on a case-by-case basis and are allowed up to 5 days after the grades are released. Requests must be created through a private post on Ed Discussion, so that it's visible to you and the staff only. Also, make sure you select "**Regrade Request**" as the category of your post. Regrade requests may take up to a week to process at the discretion of the faculty. When submitting a regrade request, please explain (in detail) why you feel the grading is incorrect.

Extra Credit

There is no extra credit in this course.

Other Course Activities

The following activities are not mandatory, but will greatly support your success on the graded assignments.

Discussion Forum

Discussion forums (on Ed Discussion) are designed to give you optional extra practice with the material and to see examples of how your classmates are thinking and working.

We strongly suggest you enroll in the Ed Discussion on the first day of class, and you can click on the Ed Discussion Tab on the navigation menu on your Canvas site.

For any course-related questions, please post them on the Ed Discussion. All the course announcements will be posted on the Ed Discussion.

Recitation

Recitations are weekly live sessions with your TAs designed around some kind of problem/activity that is supposed to take about an hour to solve. TAs will answer questions as you work through the problem or answer questions that you have submitted in advance of the session. TAs will also review any relevant quiz material from the previous week. If you are not able to attend either of the scheduled recitation times, you can review a recording of the session, posted the day after.

Additional Segments

The professor may add additional optional segments to support the class as needed.

Creating an Inclusive Environment

All members of the course community – the instructor, TAs, and students – are expected to work together to create a supportive, inclusive environment that welcomes all students, regardless of their race, ethnicity, gender identity, sexuality, religious beliefs, physical or mental health status, or socioeconomic status. Diversity, inclusion, and belonging are all core values of the MCIT Online program, the instruction staff, and this course. **All participants in this course deserve to and should expect to be treated with respect by other members of the community.**

Discussion boards, messaging channels, recitations, office hours, and group working time should be spaces where everyone feels welcome and safe. In order to facilitate a welcoming environment, students of this course are expected to:

- Exercise consideration and respect in their speech and actions
- Attempt collaboration and consideration, including listening to opposing perspectives and authentically and respectfully raising concerns, before the conflict
- Refrain from demeaning, discriminatory, or harassing behavior and speech

All members of the course community are expected to be familiar with and abide by the University's guidelines on general conduct and sexual harassment:

- University Code of Conduct: https://catalog.upenn.edu/pennbook/code-of-student-conduct/
- University Sexual Harassment Policy: http://www.upenn.edu/affirm-action/introsh.html

Students should also be familiar with other University guidelines regarding personal conduct:

- Conduct & Personal Responsibility guidelines in Pennbook: https://catalog.upenn.edu/pennbook/#policiesbytopictext
- University Principles of Responsible Conduct: http://www.upenn.edu/audit/oacp_principles.htm

If you are a victim of, witness, or are otherwise affected by unacceptable behavior:

- In cases of sexual harassment or assault, please consult DPS Special Services (https://www.publicsafety.upenn.edu/about/special-services/sensitive-crimes/ at 215-573-3333; this is a confidential resource.
- To report other bias incidents, contact the Penn Office of Diversity: https://diversity.upenn.edu/diversity-at-penn/bias-motivated-incident-report
- For other violations of the code of student conduct, the CSA Center for Community Standards and Accountability has an incident reporting form at https://csa.upenn.edu/community-standards/refer-caserequest-consultation

If you are unsure which office to contact, please contact the instructor or any Penn Engineering Online Learning staff member.

Getting and Giving Help

TA and Faculty Support

TAs will hold office hours weekly where they will open a queue in PennLab's OHQ.io system OHQ.io Resource**.** Your professor will be available for a limited number of private meetings per week, depending on the needs of the class.

Collaboration Guidelines and Academic Integrity

In the professional world of software development, collaboration—including using code that others have written—is both practical and ubiquitous. However, to prepare for entering that professional context, you need to develop a full set of software development skills so that you are both able to create your own code and evaluate the quality of someone else's code that you might use. In the context of this course, independent work and evaluation are critical. It is important to uphold academic integrity by properly acknowledging sources and giving credit where it is due.
Do not share solutions and code snippets with others on individual graded assignments unless it is explicitly indicated. Homework 7 and 8 are the only assignments in this course where students may work with another student. All other assignments are to be done individually, and code cannot be shared, even after the grades are released. Inappropriate collaboration, including sharing code without proper referencing, will be considered cheating and subject to the penalties outlined in Penn's Code of Academic Integrity.

Discussion with your peers:

> Discussion forums and recitations are collaborative—please take advantage of those times to work with your peers. Forming study groups to understand the material is a good idea, as long as you stay on the conceptual level and do not share solutions and code snippets from the graded assignments directly. Engaging in discussions with peers can deepen your understanding of the course material and foster a supportive learning environment.

Searching information Online:

> Be extremely careful when searching for information online. While it is important to independently research and broaden your knowledge, it is crucial to maintain academic integrity. Keep in mind that relying solely on online resources for solutions can hinder your learning progress, as you may obtain incorrect or

overly complex solutions or miss out on the opportunity to understand and learn from the material. **Please note that searching for solutions online is considered a violation of the course policy and may be subject to academic penalties.** Additionally, posting solutions or code to publicly accessible locations, such as StackOverflow or GitHub, is also considered a form of cheating. If you choose to use GitHub or a similar cloud-based code management system to set up a remote code repository, **it is mandatory to keep that repository private to maintain academic integrity.**

Generative AI and AI Assistants:

You may use AI-powered coding assistants like GitHub Copilot for various tasks, including coding suggestions, coding syntax corrections, refactoring, code smell detection, and boilerplate code generation. When utilizing AI assistants, it is essential to practice responsible and ethical use. If you seek assistance from AI-powered tools like ChatGPT or Google Bard, **you should always use proper referencing to ensure transparency and acknowledge the assistance received, and validate the accurate before you integrate to your own code.** Failing to reference AI-powered tools used for assistance will be considered a violation of the course's academic integrity policy. **If more than 30% of your code is generated by AI-powered tools will be considered a breach of academic integrity.**

The first instance of homework plagiarism will be handled by the instructor and may include escalation to the CSA Center for Community Standards and Accountability. Second instances of exam plagiarism will be turned over immediately to the University of Pennsylvania CSA Center for Community Standards and Accountability.

Remember, academic integrity is a fundamental value that underpins your learning and growth. By embracing a culture of honesty, proper referencing, and responsible use of AI-powered tools, you not only uphold the integrity of the course but also develop the skills necessary for your future professional endeavors. If you have any doubts or concerns about using AI assistants or any other AI-powered tools, it is recommended to seek clarification from the course staff to avoid any inadvertent academic dishonesty.

Access to Materials and Content Before and After Graduation

If you would like to retain copies of your submitted assignments, you must download them from Gradescope, Canvas, Codio, and any other platforms that you submit to during the semester in which you are taking that course.

Your SEAS account will be permanently deleted in the fall following your graduation. Once your SEAS account is deleted, we will be unable to re-grant you access to the MCIT Online Program materials and course content. Therefore, if you would like to retain the content from each course after you graduate from the program, you must download the content locally to your computer during the semester that you are in the course.

Recording Notice

Public office hours, recitations, and other live events will be recorded, used, and may be made available to class participants during the current semester as well as students who take the class in future semesters.

Private office hours will also be offered and are not recorded. Students who do not wish to attend the publicly-recorded office hour may attend the private office hours.

Fall 2023 Course Schedule and Important Dates

Dates are subject to change. Please check Ed Discussion for announcements regarding schedule changes. Note, weeks run Monday through Sunday. Homework assignments are due at 11:59 PM Eastern Time on the dates specified.

Homework Assignment	Deadline
Homework 1 : Supermarket (TA Grading)	9/5
Homework 2 : Ice Cream Stand (Autograding/TA Grading)	9/11
Homework 3 : Water Tank (Autograding/ TA Grading)	9/18
Homework 4 : Make a Website (TA Grading)	9/25
EXAM 1 : Python Exam	10/8
Homework 5 : Simple 21 (Autograding/TA Grading)	10/16
Homework 6 : Movie Trivia (Autograding/TA Grading)	10/30
Homework 7 : Battleship (Autograding/TA Grading)	11/13
Homework 8 : Student Management System (TA Grading)	11/28

EXAM 2: Java Exam	12/3

Monday, 8/28 - Sunday, 9/3	**Module 1**	**Topic(s)** Introduction to Programming and The Python Language, Variables, Conditionals, Jupyter Notebook, and IDLE
Monday, 9/4 - Sunday, 9/10	**Module 2**	**Topic(s)** Intro to Lists, Loops, and Functions
Monday, 9/11 - Sunday, 9/17	**Module 3**	**Topic(s)** More with Lists, Strings, Tuples, Sets, and PyCharm
Monday, 9/18 - Sunday, 9/24	**Module 4**	**Topic(s)** Unit Testing & Test-Driven Development
Monday, 9/25 - Sunday, 10/1	**Module 5**	**Topic(s)** Dictionaries and Files
Monday, 10/2 - Sunday, 10/8	**Module 6**	**Topic(s)** Object-Oriented Programming with Python Exam 1
Monday, 10/9 - Sunday, 10/15	**Module 7**	**Topic(s)** Introduction to Java, Classes, & Eclipse
Monday, 10/16 - Sunday, 10/22	**Module 8**	**Topic(s)** Unit Testing, Arrays, & ArrayLists
Monday, 10/23 - Sunday, 10/29	**Module 9**	**Topic(s)** Static Variables, Methods, & Polymorphism Using Overloading
Monday, 10/30 - Sunday, 11/5	**Module 10**	**Topic(s)** Inheritance, Polymorphism Using Overriding, & Access Modifiers

Monday, 11/6 - Sunday, 11/12	**Module 11**	**Topic(s)** Abstract Classes, Interfaces, & Debugging
Monday, 11/13 - Sunday, 11/19	**Module 12**	**Topic(s)** File I/O & Exceptions
Monday, 11/20 - Sunday, 11/26	**Module 13**	**Topic(s)** Collections, Maps, & Regular Expressions
Monday, 11/27 - Sunday, 12/3	**Module 14**	**Topic(s)** Exam 2