



Course Syllabus – INST 126 – Fall 2023

Introduction to Programming for Information Science

INST 126

Fall 2023

Section 0102

Course Description and Learning Outcomes

This is an introduction to programming for information science majors. This course provides a path for students with diverse backgrounds to learn computer programming. You are not expected to have any prior programming experience to take this course.

You will learn *computational thinking*: how programmers analyze problems and design solutions for those problems, and practice using this skill to solve problems. Specifically, you will learn how to implement computational solutions using the Python language, which is particularly well suited for common problems that information professionals seek to solve. These problems include data collection, analysis, and management, and developing Web applications such as search engines. This is a hands-on course – both in and out of class you will be writing, analyzing, testing, and debugging code, culminating in a project that tackles a challenging real-world problem.

Throughout the course, we will examine social and cultural issues connected to computer programming that are central to our discipline, like algorithmic bias, the distinction between ethical and unethical uses of algorithms, and disparities in access to tech jobs. You need to understand these issues to be an ethical professional and to be successful in your work, as they reflect the broader social and cultural context in which we produce software. Through assignments and class activities, we will critically examine how programming is situated in broader social and organizational structures, how it reflects these structures, the ethical and equity issues this entails, and ways that we might address these issues as information professionals.

After successfully completing this course, you will be able to:

1. Explain fundamental programming principles, concepts and methods.
2. Develop small-scale computer programs by applying fundamental programming concepts such as variables, data types, assignments, arrays, conditionals, loops, functions, and input/output operations.
3. Test and assess the quality of small-scale programs.
4. Write clear and effective in-code comments and other documentation.
5. Apply computational thinking techniques to analyze problems and develop computational solutions.
6. Explain how programming is situated in and reflects broader social and organizational structures, and the ethical and equity issues this entails.

Required Resources

- ❖ **ELMS** – The course website is at elms.umd.edu. We will use it to host the content of the course and all course-related announcements. You are responsible for staying abreast of any communications posted there.

Contact Email

TBD

(Note: This email reaches the whole instructional staff.)

Class Meets

Tue/Thu 3:30 – 4:45 PM

MCB 1207

Map: go.umd.edu/mcb1207

Instructional Staff

C. Scott Dempwolf

dempy@umd.edu

Sravya Lenka (TA)

Ya-Ting Yang (Grader)

Office Hours

Tue/Thu 5:00–6:00 PM

TBD

TA Office Hours

TBD (check ELMS)

Prerequisites

N/A

Communication

- ❖ You are responsible for checking ELMS for any announcement.
- ❖ Use ELMS for messages to instructor, and TA.
- ❖ Email subject headings should begin with “INST126 - ”.

- ❖ **Textbook** – *Python for Everybody*, by C. R. Severance. 1st ed. (2016) ISBN 1530051126. The book is freely available at: www.py4e.com/book.
- ❖ **Computer and Internet connection** – Please contact me as soon as possible if you are concerned about not having access to any of these, and I will work with you to find a reasonable arrangement. (Laptops can be borrowed at McKeldin Library, see www.lib.umd.edu/find/borrow/equipment.)
- ❖ **Python (version 3)** – To install it, go to (<https://www.python.org/downloads/>) and follow the instructions.

Campus Policies

It is our shared responsibility to know and abide by the University of Maryland’s policies that relate to all courses, including this one. These policies cover (but are not limited to) topics like:

- Academic integrity
- Student and instructor conduct
- Accessibility and accommodations
- Attendance and excused absences
- Grades and appeals
- Copyright and intellectual property

Please visit www.ugst.umd.edu/courserelatedpolicies.html for the full list of campus-wide policies.

Course overview and expectations

This is a project-oriented course, divided into four modules. At the end of each module there is a project (PR) where you create a program that solves some nontrivial problem (for example, an email parser that recognizes the various components of an email address; a document indexer for articles and books; a data analysis pipeline to analyze labor income data by race and gender). The projects will depend on your being able to write “component” code as building blocks, such as creating/updating variables, conditional blocks, iteration blocks, etc. To learn how these work, every week we release a set of programming component exercises (PCE), and weekly learning checks (quizzes) on ELMS. There are also required weekly discussion posts. There is no final exam in this course.

Assessments

Your final grade will depend on the following assessments:

Learning Assessments	How Many	Total Weight
Projects (PR)	Project 1 (5%) Project 2 (10%) Project 3 (10%) Project 4 (15%)	40%
Programming Component Exercises (PCE)	15 (approx.)	30%
Learning Checks (LCs or Quizzes)	15 (approx.)	15%
Discussions	15 (approx.)	15%

Tentative Course Schedule (monitor ELMS for current deadlines)

Session	Week / Module	Date	Day	Topic	Reading	Assignments
1	1	8/29/2023	T	Course Intro; Why Learn Programming?; Computational Thinking; Setup	Py4E Chapter 1	Quiz 1, Exercises 1, Discussion 1
2	1	8/31/2023	TH			
3	2	9/5/2023	T	Variables, expressions, and statements	Py4E Chapter 2	Quiz 2, Exercises 2, Discussion 2
4	2	9/7/2023	TH			
5	3	9/12/2023	T	Conditional Execution	Py4E Chapter 3	Quiz 3, Exercises 3, Discussion 3
6	3	9/14/2023	TH			
7	4	9/19/2023	T	Functions	Py4E Chapter 4	Quiz 4, Exercises 4, Discussion 4
8	4	9/21/2023	TH			
9	5	9/26/2023	T	Iteration	Py4E Chapter 5	Quiz 5, Exercises 5, Project 1, Discussion 5
10	5	9/28/2023	TH			
11	6	10/3/2023	T	Strings	Py4E Chapter 6	Quiz 6, Exercises 6, Discussion 6
12	6	10/5/2023	TH			
13	7	10/10/2023	T	Files	Py4E Chapter 7	Quiz 7, Exercises 7, Discussion 7
14	7	10/12/2023	TH			
15	8	10/17/2023	T	Lists	Py4E Chapter 8	Quiz 8, Exercises 8, Project 2, Discussion 8
16	8	10/19/2023	TH			
17	9	10/24/2023	T	Dictionaries	Py4E Chapter 9	Quiz 9, Exercises 9, Discussion 9
18	9	10/26/2023	TH			
19	10	10/31/2023	T	Tuples	Py4E Chapter 10	Quiz 10, Exercises 10, Discussion 10
20	10	11/2/2023	TH			
21	11	11/7/2023	T	Regular Expressions	Py4E Chapter 11	Quiz 11, Exercises 11, Discussion 11
22	11	11/9/2023	TH			
23	12	11/14/2023	T	Networked Programs	Py4E Chapter 12	Quiz 12, Exercises 12, Project 3, Discussion 12
24	12	11/16/2023	TH			
25	13	11/21/2023	T	Web Services	Py4E Chapter 13	Quiz 13, Exercises 13, Discussion 13
26	13	11/23/2023	TH			
27	14	11/28/2023	T	Object Oriented Programming	Py4E Chapter 14	Quiz 14, Exercises 14, Discussion 14
28	14	11/30/2023	TH			
29	15	12/5/2023	T	Everything Else	Py4E Chapter 15	Quiz 15, Exercises 15, Discussion 15
30	15	12/7/2023	TH			
31	16	12/12/2023	T	No Class	Py4E Chapter 16	Final Assignments Due

Grading

Some assignments and quizzes for this course will be auto graded by ELMS. Others will be graded by the instructional staff. Grades for individual assignments and quizzes are recorded in ELMS as “points”. Each assignment or quiz has a specified max number of points. To determine your grade on individual assignments, divide the points you received by the max available points for the assignment to get your grade percentage. To convert that to a letter grade, use the following table. Final grades are calculated in a similar fashion. At the end of the semester, final grades are exported directly out of ELMS to the registrar’s grade system.

A+ 97-100*	B+ 87-89.99	C+ 77-79.99	D+ 67-69.99	
A 93-96.99	B 83-86.99	C 73-76.99	D 63-66.99	
A- 90-92.99	B- 80-82.99	C- 70-72.99	D- 60-62.99	F 0-59.99

Course-Specific Policies

LATE WORK POLICY & MAKE UPS

For all types of assignments (PRs, EAs, Quizzes, and PCEs), extensions on the due date can be granted only for excused absences. This is what the official UMD policy says about the kind of events that justify an excused absence:

Events that justify an excused absence include religious observances; mandatory military obligation; illness of the student or illness of an immediate family member; participation in university activities at the request of university authorities; and compelling circumstances beyond the student’s control (e.g., death in the family, required court appearance). Absences stemming from work duties other than military obligation (e.g., unexpected changes in shift assignments) and traffic/transit problems do not typically qualify for excused absence. [Source: policies.umd.edu/student-affairs/university-of-maryland-policy-on-excused-absence]

Note that unforeseen technical problems (for example, broken laptop, WiFi down, etc.) are typically not events that justify an excused absence, unless there is a confirmed outage of university infrastructure. Keep in mind also that ELMS counts late days in blocks of 24 hours, so a submission that is late by even a few minutes will be counted as being one day late. *So, plan ahead and start working early*, so that you have plenty of time to deal with any technical problem that may prevent you from submitting on time. (For IT help, contact itsupport.umd.edu/itsupport or 301.405.1500.)

COLLABORATION, GROUP WORK, AND ACADEMIC INTEGRITY

Learning computer programming requires hard work and practice. The University has a Code of Academic Integrity (policies.umd.edu/academic-affairs/university-of-maryland-code-of-academic-integrity) about when and how you may collaborate with other students. These policies are designed to support your learning: if you subvert them, you are not getting the practice that you need to improve. All students are expected to adhere to this Code; it is your responsibility to read it and know what it says, so you can start your civic and professional life on the right path.

It is important to note that course assistance websites, including (but not limited to) Chegg or CourseHero, are not permitted sources for this and other courses, unless the instructor explicitly gives permission for you to use one of these sites. Material taken or copied from these sites can be deemed unauthorized material and a violation of

academic integrity, as well as of copyright. Furthermore, these sites offer information that might not be accurate and that shortcuts the learning process, particularly the critical thinking steps necessary for college-level assignments. The same applies to AI chatbots or code completion tools like ChatGPT, Github Copilot, or similar.

It is understandable that students may use a variety of online or virtual forums for course-wide discussion (e.g., GroupME, Discord, or WeChat). Collaboration in this way regarding concepts discussed in this course is permissible. However, collaboration on graded assignments is strictly prohibited unless otherwise stated. Examples of prohibited collaboration include asking classmates for answers on quizzes or exams, asking for access codes to clicker polls, etc. A general rule of thumb is that *exchanging or soliciting ideas about how to solve problems is not cheating, but exchanging code is cheating*. So, feel free to discuss your solutions with other students as long as you do not provide them or allow them to view your source code: if you are speaking or signing in English or another human language that's fine; if you are exchanging computer code, that's cheating.

Typical penalties for academic dishonesty can include a 0 on the assignment or an automatic failure and “XF” on your transcript. We will use anti-plagiarism software (MOSS, GPTZero) to check for potential instances of cheating.

Last Update: August 25, 2023