Course Syllabus – INST 126 – Spring 2019

Introduction to Programming for Information Professionals

INST 126
Spring 2019

Learning Outcomes

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

You will learn how programmers analyze problems and design solutions for those problems using **computational thinking**, and practice using computational thinking to solve problems. You will learn how implement computational solutions using the **Python** language (https://python.org/), which is particularly well suited for common problems that information professionals seek to solve, such as 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 issues like algorithmic bias, ethical/unethical uses of algorithms and disparities in opportunities in tech jobs, which you need to understand to be an **ethical professional** and to be successful in your work. These issues reflect the broader social and cultural context in which we produce software. The social and cultural impacts of information and technology are central concepts in our discipline. Through readings, discussion and writing, we will critically examine how programming is situated in and reflects broader social and organizational 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.

Pamela Duffy
pduffy@umd.edu

Class Meets
Tue/Thu, 2:00-3:15 pm
ARC #1101

Office Hours
See Canvas

TAs/ AMPs
(contact via ELMS)
Daniel Antwi, TA
Jonathan Bui, AMP
Tyson Nguyen, AMP

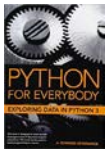
Prerequisites
MATH115 w ≥ C- or math eligibility of ≥ MATH140. Must not have completed INST326 or CMSC131. Must be in Info Sci program.

Course Communication
All course information will be posted on ELMS. Important information will be communicated via the Inbox, Announcements, and Discussion areas. Customize your alerts to ensure that you receive notifications via your preferred email account.

Communicate with the instructional team via Canvas inbox.
Email subject headings should begin with "INST126:".

Required Resources

Course website: **https://umd.instructure.com/courses/1262087**



Python for Everybody, Charles Russell Severance.
($10 print, free online) https://www.py4e.com/book;
Also available as an online textbook with embedded mini-editors ("trinkets")
at https://books.trinket.io/pfe/

1st edition (2016).
ISBN #1530051126

You need to download and install "Turning Point."
Physical clickers are NOT required; you MAY use a phone app.
Visit the Students section of clickers.umd.edu for details.

Laptop - We will do live programming exercises during most classes, so bring your laptop and be prepared to write code. Any reasonably current operating system can be used. If you don't have access to a laptop, contact me before the first class.

Python - Python programming language (version 3). The Anaconda programming platform which includes python and other important packages is freely available from https://www.anaconda.com/download/. Consider also following the setup instructions from the py4e website: https://www.py4e.com/lessons/install.

Optional resources:

- PyCharm Community Edition: https://www.jetbrains.com/pycharm/download/, a freely available text editor and development environment for Windows, Mac, and Linux.
- Python Language Reference: http://docs.python.org/3.6/reference/index.html
- *Think Python: How to Think Like a Computer Scientist, 2nd Ed.* by Allen Downey
  http://greenteapress.com/thinkpython2/
- Learn Python the Hard Way: https://learnpythonthehardway.org/book/ - a free book that explains all the details of Python
- PythonTutor: http://pythontutor.com/ - free tool to visually understanding what is going on in your code.
- **Free online classes:**
  - Udacity CS101: https://www.udacity.com/course/cs101
    Coursera Programming for Everybody:https://www.coursera.org/course/pythonlearn
- **Libraries:**
  - Official Python Library Reference: https://docs.python.org/3.6/library/index.html
  - Python Cheat Sheet:
    https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf
- **Other:**
  - StackOverflow programmers Q&A site: https://stackoverflow.com/ (usually works well to search via Google first, and check for related content from this site in the search results)

Campus Policies

It is our shared responsibility to know and abide by the University of Maryland's policies that relate to all courses, which include 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 Office of Undergraduate Studies' full list of campus-wide policies and follow up with me if you have questions.

Course overview and expectations

Each week will typically follow this pattern, with some exceptions (e.g., for exam weeks):

BEFORE THE WEEK BEGINS (PREPARATION)

- Do assigned readings, working each example in the text;
- Watch any assigned videos;
- Do any pre-class activities – these help you confirm that you understand the basic material and/or help you identify specific aspects that you have questions about. These pre-class activities will sometimes include callbacks to previous material to reinforce understanding;
- Your pre-class activities culminate in completion of a worksheet (on ELMS, unless otherwise specified), which is due at the beginning of each week. The worksheet helps you confirm you are prepared for class and identify any gaps or questions. I review these before class and use them to prepare for class.

DURING CLASS

We will use a mix of lecture, discussion and lots of hands-on activities to help you apply the materials. We will make extensive use of paired and group work in class, including a substantial amount of **paired programming** (a best practice in software engineering and programming education).

Class is not a time for solo learning. As members of a learning community, we are mutually responsible to each other as learners. Each of us has to be fully engaged with each other in the activities. We have to be supportive of each other as we try to explain or demonstrate something new, as we inevitably make mistakes. We aren't successful unless everyone is learning.

Our time together in class is precious. To use it effectively, you must come to class on time and prepared. Being prepared for class means that you have:

1. (Before the week begins) Completed all the readings/videos, and attempted all the pre-class activities and either successfully completed them or submitted your questions the day before class, so I have time to prepare and answer them in class.
2. (At each class) Arrive 5 minutes before class starts; are in your seat, with Python tools ready. You have downloaded any notes or materials for the day from ELMS. Any paper assignments are ready to hand in. You are ready to take notes.

AFTER CLASS:

To provide further practice and opportunities for feedback, and integrate concepts/skills across classes, you will work on homework assignments and your team project.

We will use ELMS Discussions as a forum that you can use to ask/answer questions, get clarifications, point out my mistakes, etc. Be sure to check it regularly.

Here is my suggested general strategy for working on assignments:

1. Start early – don't wait. That will give you time to work through the problems and get help as needed.
2. When you run into a problem, spend 5-10 minutes trying to solve it on your own.
3. Then take a break. Sometimes this will allow you to come back and see something you missed. Letting your sub-conscious work on it for a while (unsupervised, so to speak) will often lead to useful ideas.
4. If you've spent 20-30 minutes and still are stuck, post your question online. We are here to help each other, so don't beat your head against a brick wall - ask for help! When you post, provide as much information as you can. Often it helps to post a screenshot with the problem.
5. The instructional team will monitor and will respond as soon as we are able, usually within a day (longer during weekends, travel, etc.). If I don't respond, you can expect responses (usually within a day) from either the TA or AMPs (your peer mentors in the class) during weekdays.
6. If you see a question on the Discussions that you can answer, or if you have an idea, please respond. Don't wait for me. You will be helping your colleagues.

Assessments

1: WORKSHEETS (12% OF FINAL GRADE)

These worksheets help you assess your preparation for the concepts to be discussed in class. They will typically be quizzes submitted on ELMS before the first lecture of the week, consisting of a mixture of multiple choice, matching, and short answers. They are designed to be straightforward to do well on as long as you have done the reading. The lowest grade is dropped.

These are due by 11:59 pm on the Sunday before the first lecture of this week (except for Week 1), so that I can take this into account for planning for the week's lecture (e.g., if there are particular concepts with which many students are struggling). These will generally be opened on ELMS at least 1 week before they are due. Please let me know if you need earlier access to a worksheet (e.g., to complete your work early in advance of pre-approved travel; see also policy on missed deadlines below).

2: IN-CLASS LEARNING CHECKS (5%)

We will use learning checks in the first session of the week (mostly with clicker questions) to make sure that you are following along with the content. They should be straightforward and you can do well if you are paying attention in class and have done the reading. The lowest grade is dropped: consider this a "freebie" for when unexpected life circumstances get in the way (e.g., you get stuck in traffic or have an unexpected emergency to attend; see policy on Missed Deadlines below for what to do if you know ahead of time that life will get in the way).

3: IN-CLASS EXERCISES (12% OF FINAL GRADE)

Each week, a substantial portion of the last class will be devoted to a hands-on exercise that you will turn in to be graded. These will typically be hands-on programing exercises that apply the concepts and skills discussed in class, the vast majority of which will be completed in pairs (for pair programming). Sometimes these include discussion exercises as well. "Correctness" of your programs will be an important component of your grade, but you should have sufficient support to do well on them in class. Similar to the learning checks, the lowest grade is dropped.

4: HOMEWORKS (25% OF FINAL GRADE)

Includes coding problems (of course), but will also include analysis questions, brief reflective writing, and other activities. For these assessments, "correctness" of your solutions will be an important part of your grade. There will be 5 of these assignments.

5: EXAMS (25% OF FINAL GRADE)

These are diagnostics for you to assess your understanding of the programming basics that are necessary as you proceed through the course and prepare for term project. You will want to address any weaknesses these diagnostics identify to ensure you are well prepared for the project.

There will be 2 exams: the first exam covers material from Weeks 1-5; the second exam covers material from Weeks 6-10. Each exam will consist of 2 parts: 1) an in-class (on paper), closed book exam, and 2) a take-home exam consisting of coding problems, released the day of the in-class exam and due 4 days after the in-class portion.

6: TEAM PROJECT (16% OF FINAL GRADE)

The team project will give you an opportunity to apply what you learn in class (especially computational thinking practices and testing/documentation best practices) in a more realistic application than toy problems. You will form a small (~4-5 people) team to **formulate and develop a solution to a real-world problem that interests you** (some samples will be provided). The exact problem you choose is open-ended, provided it conforms to at least one of three application areas that are relevant to information science: data analysis, search, or web services.

There will be **5** components to the team project:

- **1** initial project proposal: Your team will submit a document that describes a formulation of your problem in computational terms, along with your team's collaboration plan (proposed roles, time/scheduling expectations). Worth 5% of your final grade for the course.
- **3** progress updates (including current state of the code): Each update is worth 2% of your final grade for the course, for a total of 6%.
- **1** final project submission: worth 5% of your final grade for the course.

Assessment on this project will be process-centric: your grade will heavily focus on assessing the extent to which you are effectively using computational thinking techniques, including computationally formulating problems, using decomposition and abstraction, and practicing good software engineering techniques like quality testing (e.g., testing for edge cases), and effective documentation. It is possible that some of you may choose problems that turn out to be too difficult to solve (although we will work hard to make sure your chosen problems are scoped well enough that you have a good chance of succeeding). If this happens, don't worry! As long as you are doing a good job of following good *processes*, you should be able to do well on this assessment.
I also recognize that social loafing can be a problem: to incentivize everyone to pull their weight, I will require you to 1) submit a collaboration plan with your initial proposal, 2) come to at least 2 of the 3 planned in-class project work sessions (your grade on the team project will be reduced by 10% for missing 1, and 20% for missing 2), and 3) complete (confidential) peer reviews of your teammates.

Grading

Grades are earned, not given. Your grade is determined by your performance on the learning assessments in the course. If earning a particular grade is important to you, please speak with me at the beginning of the semester so that I can offer some helpful suggestions for achieving your goal.

Assessment scores will be posted on the course ELMS page. If you would like to discuss your grade or have questions about how something was scored, please schedule a time with the course TA. Grade disputes must be submitted in writing within one week of receiving the grade.

This table illustrates the percentage weight of each assessment towards your final grade.

Learning Assessments	Percentage
Worksheets (WS): ~13, lowest grade dropped	12%
In-Class Learning Checks (LC): ~11, lowest grade dropped	5%
In-Class Exercises (IC): ~13, lowest grade dropped	12%
Homework (HW): 5	25%
Team Project Project Proposal Updates: 3 Final Submission	16%
Exams (EX): 2	25%

Letter grades will be assigned using the following scale.

Grading Scale											
A+	≥ 97%	B+	≥ 87.00%	C+	≥ 77.00%	D+	≥ 67.00%				
A	≥ 93.00%	B	≥ 83.00%	C	≥ 73.00%	D	≥ 63.00%	F	<60.0%		
A-	≥ 90.00%	B-	≥ 80.00%	C-	≥ 70.00%	D-	≥ 60.00%				

Note: There is no rounding: for example, an A- is at least 90.00, not 89.5 or 89.9. **I will not respond to email requests for a grade bump at the end of the semester.**

Course-Specific Policies

Devices in class

I expect you to make the responsible and respectful decision to refrain from using your cellphone in class except as a clicker. If you have critical communication to attend to, please excuse yourself and return when you are ready. For more information about the science behind the policy watch: http://youtu.be/WwPaw3Fx5Hk

Missed Deadlines

If you will not be able to meet an assignment deadline, contact the instructor **before the due date** to explain why you will need to submit the assignment late and what your plan is; these will be evaluated on a case-by-case basis.

Unless prior permission has been granted, **no late work is accepted**. This policy is in place to ensure all students have their work returned to them in a timely fashion. Please prepare in advance so that you will not encounter technical difficulties that may prevent submission of a given assignment. If you have a conflict with the due date, assignments can always be submitted early. Generally speaking, illnesses are not an excuse for late assignments because you will receive the assignments at least one week before they are due.

> ***Note*: Exams are not included in the missed deadlines policy. See next section.**

Exam Policy

If you need to miss an exam because of outside circumstances (e.g., a religious holiday, military duties, work/athletic team travel), you must email me **before the exam** to reschedule your exam time. If you are sick on an exam day, you must provide me with a doctor's note to be excused (see the UMD policies on absences) and should email me before the exam time to let me know you're sick. If you miss an exam due to other circumstances (e.g., oversleeping), you will not be able to make up the exam.

Collaboration, Group Work, and Academic Integrity

All of the individually graded assessments must be completed independently. You are welcome (and highly encouraged) to study and discuss the course material with your peers, but providing or receiving quiz/exam answers or letting someone else contribute to your writing assignment constitutes academic dishonesty. Penalties for academic dishonesty can include a 0 on the assignment or an automatic failure and "XF" on your transcript.

For the team project assignments, you may and should collaborate with members of your team (but not other teams). To address social loafing, **you will be asked to complete confidential individual peer reviews of your teammates**—team members who don't pull their weight may receive a different grade than the team.

The Worksheets (WS) are open-book. This means that you may consult the readings or your notes (but not another person) as you take the quiz.

The in-class portion of each **Exam (EX) is closed-book.** The take-home coding problems portion is open-book. This means you may consult the readings or your notes (but not another person) to complete the coding problems.

Get Some Help!

You are expected to take personal responsibility for you own learning. This includes acknowledging when your performance does not match your goals and doing something about it. Everyone can benefit from some expert guidance on time management, note taking, and exam preparation, so I encourage you to consider visiting http://ter.ps/learn and schedule an appointment with an academic coach. Sharpen your communication skills (and improve your grade) by visiting http://ter.ps/writing and schedule an appointment with the campus Writing Center. Finally, if you just need someone to talk to, visit http://www.counseling.umd.edu.

Everything is free because you have already paid for it, and **everyone needs help**… all you have to do is ask for it.

Names/Pronouns and Self Identifications

The University of Maryland recognizes the importance of a diverse student body, and we are committed to fostering equitable classroom environments. I invite you, if you wish, to tell us how you want to be referred to both in terms of your name and your pronouns (he/him, she/her, they/them, etc.). The pronouns someone indicates are not necessarily indicative of their gender identity. Visit trans.umd.edu to learn more.

Additionally, how you identify in terms of your gender, race, class, sexuality, religion, and dis/ability, among all aspects of your identity, is your choice whether to disclose (e.g., should it come up in classroom conversation about our experiences and perspectives) and should be self-identified, not presumed or imposed. I will do my best to address and refer to all students accordingly, and I ask you to do the same for all of your fellow Terps.

Course Schedule

WS = Worksheet submitted online by **11:59pm** on Sunday that week (except Week 1). These will generally be opened on ELMS at least 1 week before they are due.
HW = Homework submitted online by **11:59pm** on Thursday of the week
Note: No assignment will be accepted for credit after the deadline.

IC = Hands-on In-Class Exercises
LC = In-Class Learning Checks
TP = Team Project Assignment

Week	Topic	Due at beginning of week	Date	During our Class Meeting	Other due dates
1	Intro to computational thinking	Read PY4E Ch1 Complete **WS-01** (due Wed)	Tue 1/29	WEATHER – CLASS CANCELLED	
			Thu 1/31	Course overview and start on coursework	
2	Catch-up and Basic program components and structure	Read PY4E Ch2-3 Complete **WS-02**	Tue 2/5	What is computational thinking? What is Python?	*Must have TurningPoint setup by Thursday of this week*
			Thu 2/7	Variables and conditionals **LC-01 and IC-01**	
3	Basic program components and structure, ctd.	Read PY4E Ch4-5 Complete **WS-03**	Tue 2/12	Functions and Iteration **LC-02**	
			Thu 2/14	**IC-02**	
4	Data structures: Strings	Read PY4E Ch6 Complete **WS-04**	Tue 2/19	Strings **LC-03**	Mon: **HW-01:** Variables, Conditionals, and Computational Thinking
			Thu 2/21	**IC-03**	
5	Data structures: Lists	Read PY4E Ch8 Complete **WS-05**	Tue 2/26	Lists **LC-04**	Fri: **HW-02:** Strings
			Thu 2/28	**IC-04**	
6	Program external memory	Read PY4E Ch7 Complete **WS-06**	Tue 3/5	Files **IC-05**	
			Thu 3/7	**Exam 1** (in class)	
7	Advanced data structures: Dictionaries	Read PY4E Ch9 Complete **WS-07**	Tue 3/12	Dictionaries. **LC-06**	Mon: **Exam 1** take-home
			Thu 3/14	**IC-06**	
8	**Spring Break**				
9	Applications: Search	Complete **WS-08**	Tue 3/26	Search engines **LC-07**	Fri: **HW-03:** Lists and Dictionaries
			Thu 3/28	**IC-07**	
10	Applications: Search	Complete **WS-09**	Tue 4/2	Dictionaries review	Fri: **TP-01** (Project Proposal)
			Thu 4/4	Project kickoff **IC-08**	
11	Applications: Data Analysis	Complete **WS-10**	Tue 4/9	Review dictionaries, working with larger programs **LC-08**	Fri: **TP-02** (Update 1)
			Thu 4/11	Brief intro to Pandas **IC-09**	

12	Applications: Data Analysis, ctd.		Tue 4/16	**Exam 2** (in class)	Sat: **Exam 2** take-home
			Thu 4/18	Pandas **LC-10 IC-10**	
13	Applications: Data Analysis, ctd.	Complete **WS-11**	Tue 4/23	Pandas/Matplotlib **LC-11**	Fri: **HW-04:** Data Analysis
			Thu 4/25	**IC-11**	
14	Applications: Web Services	Read "Don't Get Distracted" (TBP on ELMS) Read PY4E Ch12 (focus on these sections: 1) "Retrieving web pages with urllib", and "Parsing HTML using BeautifulSoup")	Tue 4/30	Project work / ethics in programming **IC-12**	Tue: **TP-03** (Update 2)
			Thu 5/2	Web Services **IC-13**	
15	Applications: Web Services, ctd.	Read PY4E Ch13 Complete **WS-13**	Tue 5/7	Web Services 2 **IC-14**	Tue: **TP-04** (Update 3) Fri: **HW-05:** Web services
			Thu 5/9	**Project work**	
16	Project Completions		Tue 5/14	Project demos! **IC-15**	
	No final exam - The final project report is used instead of a final exam. **All materials are due by Friday 5/17 at 10:00am.**				Fri: **TP-05** (Final project deliverables and report)

Note: This is a tentative schedule, and subject to change as necessary – monitor the course ELMS page for current deadlines. In the unlikely event of a prolonged university closing, or an extended absence from the university, adjustments to the course schedule, deadlines, and assignments will be made based on the duration of the closing and the specific dates missed.