UNIVERSITY OF MICHIGAN
SI 106: Programs, Information, and People
Release 1.0, as of 12/24/2015

Term:	Winter 2016
Lectures:	MW 2:30-4:00
Location:	NatSci Auditorium
Instructors:	Prof. Paul Resnick Jackie Cohen
GSIs:	Tera Reynolds Sam Carton Chanda Phelan
IAs:	Ayomide (Ayo) Akinokun Lauren Murphy Natalie Wysocki Salvatore (Sal) Digioia

Office Hours

Prof. Resnick Individual appointments (private consultations)	Fridays 1:30-2:00 in 4322 NQ; self-schedule for 10-minute slots via this URL
Jackie Cohen (individual appts/private consultations)	Wednesdays 4:15-5:05 (after lecture); 1270 NQ; self-schedule for 10-min slots via this URL
Team; group office hours (homework help; exam prep)	Fridays 2-6 Sunday 1-3 2185 NQ

Instructional team email list: **instructors-si106w16@umich.edu** will reach the entire instructional team. Use it when there's a problem with grading, or other administrative matters. For substantive questions, post on the Facebook group https://www.facebook.com/groups/1752935254934382/

Course Description

Introduction to programming with a focus on applications in informatics. Covers the fundamental elements of a modern programming language and how to access data on the Internet. Explores how humans and technology complement one another, including techniques used to coordinate groups of people working together on software development.

Where this course fits in the curriculum

This course is the first in a two-course sequence introducing students to programming and the culture of programming, with a focus on applications *for people, by people, and about people*. For people means applications for end-users or analysts, as opposed to back-end or infrastructure software. By and about people refers to processing data traces of people's actions and interactions.

Terminology, for skill-oriented learning objectives:

- **mastery**: could do a task requiring this skill at the same level of skill as people who are paid full-time to do this kind of work; could teach this skill to others; could knowledgeably answer questions about this in an on-line forum and get upvoted by other participants.
- **competency**: could do a task requiring this skill on their own without delegating it, though not with the fluency of someone who had mastery; could post a well-formed question in an on-line forum that elicited a serious answer from someone with mastery.
- **literacy**: could play a meaningful role in a project where someone else exercises this skill and would understand what the other person was doing; could search in an on-line forum for an answer to a question, and interpret the response.
- **awareness**: knows that this domain of knowledge exists; could fearlessly start the task of learning more through self-study or additional coursework.

High Level Objectives for this Course

- mastery: none
- competency
  - install, configure, and use a programming environment
  - read, parse, process and reformat text files using python
- literacy
  - scrape data from external sources using a web crawler or an API
- awareness
  - contribute to an open-source software project
  - create test suites

Detailed Learning Objectives

At the end of this course, students should be able to:

- Discuss the ways that people and computers are the same and ways that they are different as information processors
- Describe how open source software projects are typically organized and some of the advantages and disadvantages of that organizational form
- Describe the relationship between redundancy and compression
- Use a UNIX command prompt and the following commands, features, and utilities:
  - cd
  - ls
  - chmod
  - cat
  - less
  - grep
  - <, >, and |
- Understand the following programming concepts:
  - Data types
  - Variables

    - Functions
    - Conditional statements
    - Iteration
    - List and dictionary data structures
    - List comprehensions
    - APIs
    - Classes and inheritance
- Write programs in python that demonstrate understanding of all of the above concepts and that use the following features:
    - File operations
    - String processing operations
    - External modules and APIs
- Manipulate data to
    - Extract and summarize desired elements
    - Output the processed data in .csv format
- Create test cases for a simple program

Class Structure

We will be using a partially “flipped” classroom. It is expected that you will read and attempt exercises before class. To do this well, you will generally need to allocate about 1-2 hours of prep time before each lecture session. You will also need to allocate about 1 hour of reading time each week for the non-programming readings.

During the official lecture time, we will review tricky points and difficult exercises, solve some additional problems together, discuss the non-programming readings (Most Human Human; Success of Open Source; etc.), and get a preview of the next session’s material. We will also go over solutions to some parts of problem sets.

During the section meetings, you will work on additional problems together and have supervised time working individually on the graded homework.

If you are doing the 4 hours of prep time weekly, you will often be able to complete the graded homeworks with 1-4 additional hours outside of class. If you don’t do the prep before class, you will lose much of the benefit of the class time, and will end up spending more total time trying to catch up in order to do the homework. You will also lose the points for prep, so it will hurt your grade directly.

Required Resources

Textbooks and Notes

We will be using a custom version of an interactive, free, online textbook.

Programs, Information, and People

Paul Resnick

Access it by logging into Canvas (the replacement for cTools), clicking on Assignments and then "Textbook login".

The content for the book has a long history. Another "fork" of it can be found online at www.pythonlearn.com/book.php.
You can also download a PDF of that version.

Other Required Readings

Christian, B., The Most Human Human, 2011, London, England: Viking.
Weber, S. The Success of Open Source, 2005, Cambridge, MA: Harvard University Press

You will need to buy those two books or otherwise acquire them!

Some other articles and news stories as I find them; these will be available online.

Course Q&A Site

The best way to get help during the course will be via the class Facebook group, https://www.facebook.com/groups/1752935254934382/. You are encouraged to respond to (comment on) other students' messages. You are also encouraged to "like" questions, answers, or comments that you find helpful. Don't worry about giving wrong information: the instructors will post corrections if necessary. We will also be downloading the posts, comments, and likes data from that page to use as data for you to analyze in some of your homeworks: you'll be able to count who posts the most and who gets the most likes, for example.

If you have something of a private nature that you don't want to share with the other students, please send a message to the instructors' email address (instructors-si106w16@umich.edu) rather than posting it on the Facebook group. If it's something you don't want to share with the whole instructional team, you can email an individual instructor or talk to us after class or at office hours. Please do not use Facebook messenger for communication with the instructional staff. Please send all grading concerns to the entire instructional team via the email list barring special circumstances of privacy.

Academic Integrity

Unless otherwise specified in an assignment all submitted work must be your own, original work. Any excerpts, statements, or phrases from the work of others, including code snippets, must be clearly identified as a quotation, and a proper citation provided. Generally, you will be permitted to work with others on the weekly problem sets (more details below) with the exception of one final problem which you will be required to do entirely on your own.

Any violation of the School's policy on Academic and Professional Integrity (stated in the Master's and Doctoral Student Handbooks) will result in serious penalties, which might range from failing an assign-ment, to failing a course, to being expelled from the program.

Violations of academic and professional integrity will be reported to UMSI Student Affairs. Consequences impacting assignment or course grades are determined by the faculty instructor; additional sanctions may be imposed by the Assistant Dean for Academic and Student Affairs.

Giving and Receiving Assistance

The first time you learn technical material it is often challenging. We are going to cover a wide range of topics in the course and we will move quickly between topics. Because it is my goal for you to succeed in the course, I encourage you to get help from anyone you like.

However, you are responsible for learning the material, and you should make sure that all of the assistance you are getting is focused on gaining knowledge, not just on getting through the assignments. If you receive too much help and/or fail to master the material, you will crash and burn later in the semester. The final submission of each homework exercise must be in your own words.

If you receive assistance on an assignment, please indicate the nature and the amount of assistance you received. If the assignment is computer code, add a comment indicating who helped you and how. Any excerpts from the work of others must be clearly identified as a quotation, and a proper citation provided (e.g., in the comments of the code if it is a code fragment you have borrowed). If you are a more advanced student and are willing to help other students, please feel free to do so. Just remember that your goal is to help teach the material to the student receiving the help.

As a rule of thumb, if you work through a problem with others, generating the same code on each of your screens, when you get to the end of the problem set, you should be able to do both of the following:

- Explain how all of the code works to an instructor;
- Throw away your answer and reproduce it on your own, without any assistance from other students, in less than an hour.

Each week there will also be a free-form question to demonstrate your understanding of the material on the problem set. Your answer will have to go beyond what was specified in the problem set in some way, by explaining some subtlety or extending the solutions in some way. For this question, you must work individually and not in groups.

It is acceptable for this class to ask for and provide help on an assignment via the Facebook Q&A site, ***including posting code snippets***. Please don't post complete answers, though. If it seems like you've posted too much, one of the instructional staff will contact you to let you know, so don't worry about it. When in doubt, err on the side of helping your fellow students. When you post requests for help, the instructional staff will coach you, privately or publicly, on how to ask a question the right way. It takes some work to post good technical help questions and often, in the process, you will figure out the answer yourself. That's a good outcome. If you do the work to post good questions, you will find it's amazing how generous people will be with their time and expertise, both in this class and later in the wider world. We'll be looking at some examples of this on StackOverflow and other sites as the semester goes on.

After a few weeks, you will notice that some students are consistently providing good responses to others' posts in the FB group. **Please do not contact those students privately via email or FB message to ask for private help.** They are being generous enough by providing assistance in the public channel, where it does the most good.

You may find that a study group that starts out early in the semester as a mutual aid society turns more one-sided as the semester goes on. **If you find that you're always the one giving help and never asking for it, or vice-versa, please break up the study group and find others to work with.** It will be to the long-term advantage of everyone involved—you can really fall behind in understanding the material in this course if you're getting too much help on problem sets week after week, and it can be hard to recover.

To reiterate, the collaboration policy is as follows. Collaboration in the class is allowed (and even encouraged) for problem sets – you can get help from anyone as long as it is clearly acknowledged. Collaboration or outside help is not allowed on exams or reading responses, though you will be allowed to use some materials that you bring with you to exams. **Use of solutions from previous semesters is not allowed.** The authorship of any assignments must be in your own style and done by you, even if you get help. Any significant help must be acknowledged in writing.

Student Mental Health and Wellbeing

The University of Michigan is committed to advancing the mental health and wellbeing of its students, while acknowledging that a variety of issues, such as strained relationships, increased anxiety, alcohol/drug problems, and depression, directly impacts students' academic performance.

If you or someone you know is feeling overwhelmed, depressed, and/or in need of support, services are available. For help, contact Counseling and Psychological Services (CAPS) at (734) 764-8312 and https://caps.umich.edu/ during and after hours, on weekends and holidays or through its counselors physically located in schools on both North and Central Campus. You may also consult University Health Service (UHS) at (732) 764-8320 and https://www.uhs.umich.edu/mentalhealthsvcs, or for alcohol or drug concerns, see www.uhs.umich.edu/aodresources.

For a more comprehensive listing of the broad range of mental health services available on campus, please visit: http://umich.edu/~mhealth/"

Accommodations for students with disabilities

If you think you need an accommodation for a disability, please let the instructors know at your earliest convenience. Some aspects of this course, the assignments, the in-class activities, and the way we teach may be modified to facilitate your participation and progress. As soon as you make us aware of your needs, we can work with the Office of Services for Students with Disabilities (SSD) to help us determine appropriate accommodations. SSD (734-763-3000; ssd.umich.edu/) typically rec-ommends accommodations through a Verified Individualized Services and Accommodations (VISA) form. I will treat any information that you provide in as confidential a manner as possible.

Work in the Course – Getting a Grade

Assignments

There will be assignments throughout the course (pretty much every week and sometimes mini exercises during a lecture). Regular assignments allow you to learn the material in small "chunks" and to keep a close eye on how well you understand the material. In some cases, we will do part or all of the graded assignments during a lecture or discussion section.

In particular, each week you will have the following graded activities:

- **Lecture prep.** The online textbook tracks how much time you spend on each chapter and which code snippets you run and which multiple choice questions you answer. If you have enough activity in the textbook before the class meets, you'll earn the checkmark for that class session, worth 50 points. There will be 22 or 23 sessions when you have an opportunity to earn a check mark. 20 checkmarks will earn you the maximum score, so you can miss preparation for a couple sessions due to illness or whatever without penalty; there is no need to document illnesses or other special circumstances for this. Please don't ask for extras beyond those automatically available to everyone.
- **Responses to readings.** Each is a short response to the book chapters or other readings. They are due by midnight the day before the first class session where they might be discussed, and they are submitted via Canvas. Instructors will review them in preparation for class. You earn a checkmark if you make a good-faith effort on this, 0 if not. There will be 12 of these. 10 checkmarks earns full credit, so you can miss preparation for a couple sessions due to illness or whatever without penalty; there is no need to document illnesses or other special circumstances for this. Please don't ask for extras beyond those automatically available to everyone.
- **Discussion section participation.** If you attend and make a good-faith effort to participate, you will earn 100 points for that session. There are 13 discussion section meetings. 10 checkmarks earns full credit, so you can miss a couple sessions due to illness or whatever without penalty; there is no need to document illnesses or other special circumstances for this. Please don't ask for extras beyond those automatically available to everyone.
- **Problem sets.** These consist of one to several programming exercises to complete. Problem sets are due on Sundays at 5PM. Generally, if you are keeping up with class prep, you should be able to complete them on Thursday or Friday. Help is available from office hours on Friday and Sunday afternoons and online via the Facebook group through the weekend. Each problem set is worth 1000 points. There will be 12 of them. Your top 10 scores will count, so, without penalty, you can miss or do poorly on at least one problem set, due to illness or whatever; there is no need to document illnesses or other special circumstances for this. Please do not ask for additional special treatment beyond dropping your two lowest scores.

- **Demonstrate Your Understanding.** Each week, 100 points of credit can be earned for demonstrating your understanding of some of the material in class that week. These can take the form of written text, code, or another file type if you believe that a diagram or other form of work clearly demonstrates your understanding of some of that week's material. These will be submitted via Canvas, and are due an hour after the problem set is due. They are **a solo exercise: you may not work on these with other students**. You will receive feedback from an instructor on each Demonstrate Your Understanding exercise that you submit. Each week, you will receive initial feedback before the following discussion section. At that point, if you do not receive credit, you may continue to submit follow-up comments on that DMU assignment until you receive full credit, *as long as you resubmit it within two days of the last instructor comment*. More than 2 days after an instructor provided feedback, comments for that DMU assignment are not allowed. There are 12 opportunities to do this in the semester; 10 DMU submissions earns full credit. There is no need to document special circumstances for skipping one or two. Please do not ask for additional special treatment beyond dropping your two lowest scores.

Final Project

The final project will give you an opportunity to use your creativity and problem solving skills to significantly expand on one of the earlier assignments. For example, you can:

- Collect data from multiple sources on the Internet, analyze it, and present it in interesting charts.
- Create an interactive game.

Exams

There will be a midterm and a final exam. The midterm exam is administered during the regular lecture session and the final during the final exam period. The exam dates are announced now (see the dates at the end of this document). **Please note: wanting to leave town early for break or at the end of the semester is not a valid conflict for** allow you to stay in town for the final examination period or for the period when the midterm exam is given, then this is not the right semester for you to take this course. You will find that I am pretty flexible about many things in this course, but this isn't one of them. I will, of course, make accommodations for religious observance and other excused absences per university policy.

There will be a second-chance midterm about two weeks after the regular midterm. If you choose to take this exam, it can replace your score on the midterm, but you will keep only 90% of the points that you get from your exam answers. We will keep your highest score of the two (the first midterm score OR 0.9 * the second chance midterm score, whichever is larger).

You will be able to bring one 2-sided 8.5 x 11 page of notes to each exam.

Helping Others (Bonus Points)

Helping others to learn, by answering others' questions or sharing materials you've found useful, may earn bonus points. We monitor the Facebook group and office hours for this behavior and award it at the end of the semester.

Grading Summary

A total of 27,000 points are available, plus some bonus points.

Weekly problem sets (12 at 1,000 points; best 10 scores count)	10,000
Final project	2,000
Class prep (programming readings)	1,000
Class prep (non-programming readings)	1,000
Discussion section participation (13 at 100 points each; best 10 scores count)	1,000
Demonstrate Your Understanding (12 at 100 points; best 10 scores count)	1,000
Midterm exam	4,000
Final exam	7,000
Helping other students: (helpful questions or answers on Facebook group or office hours)	250

Grades will be awarded as follows:

A+	25,500
A	24,500
A-	23,500
B+	22,750
B	21,750
B-	20,750
C+	20,000
C	19,000
C-	17,500
D	14,500
F	<14,500

I target a median grade of B in the course. Note, however, that grades are not assigned on a curve. You can all earn As. So feel free to help your classmates learn and succeed in the course!

Exam scores are generally much lower than grades on other work. Except for the exams, with enough effort you should be able to get close to 100% of the points. The exams require mastery under pressure, without the benefit of a python interpreter or Internet resources, so scores tend to be lower. Median scores on exams tend to be between 60% and 80% of the available points. If you get all 15,000 non-exam points and 84% on the exams, you can earn an A. With just 51% on the exams you can earn a B-, if you earn all the non-exam points. Of course, doing the hard work to get all the non-exam points will usually lead you to doing better on the exams as well.

Course Outline

Success in the Course

The course is designed for students with no programming experience. If you stick with the course and invest the necessary time, you will be amazed at how much you will learn in 14 weeks. Don't be intimidated by the few students who already know how to program. Some students who have previously taken a programming course but didn't feel they mastered the concepts take this course. They have an easier time than those who have never programmed before. But every semester there are students who have never programmed before who earn A or even A+ grades in the course. Computer science majors who just want to learn python are strongly discouraged from taking this course. If you're a CS major, you can probably learn python pretty quickly on your own without taking this course.
The instructional staff is committed to helping every student get to the point where they are comfortable writing computer programs. The world looks like a very different place once you can do that, and we look forward to welcoming you into the club.

If you do not have any programming experience, some concepts will take some time to sink in. The assignments are the best way to track your progress through the material. If the assignments are difficult for you, I encourage you to go back and do the previous week's assignment a second time without looking at your previous answers—it will go much faster, build your confidence, and prepare you to handle the next week's assignment.

Usually the biggest problem students encounter in this course is trying to solve problem sets entirely by trial and error. You can be moderately successful early in the course without fully understanding what you've done, but that will provide you a poor foundation for later in the course. Especially early in the course, don't settle for just getting by; stick with things until you feel like you've mastered them.

Cramming does not work well when dealing with the material in this course. Everything builds on the previous material, and it should make sense by the third or fourth time you've seen it, even if it doesn't come to you immediately. If you try to learn it all at once, you are unlikely to succeed.

Good luck and welcome aboard!

Course Schedule and Important Dates

Session	DATES	TOPIC/BOOK SECTION	READING	Problem Set (due the Sunday after the class session)
1	1/6 Wed	Introduction to the Course and Computers. Quick overview of computing and Python.		
2	1/11 Mon	Variables, data types, expressions, assignment statements; function calls; print; The Turing Test.	Most Human Human: Prologue and Chapter 1.	
3	1/13 Wed	object instances; raw_input Unix command line: cd, ls, and pwd	Searle's Chinese Room argument	PS1
	1/18	NO CLASS—MLK Holiday		
4	1/20 Wed	Strings and lists; Unix command line: cat, less	Most Human Human: Chapter 2	PS2
5	1/25 Mon	Iteration on sequences; accumulation pattern;		

6	1/27 Wed	File operations;
		Conditionals;
			Most Human Human: Chapter 3	PS3

7	2/1 Mon	Dictionary data structure; Unix command line: echo, pipes, <, and >, grep		
8	2/3 Wed	Max loop pattern; Parsing a text line;	Most Human Human: Chapter 5	PS4
9	2/8 Mon	Defining functions; local and global variables;		

10	2/10 Wed	Indefinite iteration;
		Debugging techniques;
		native python interpreter;
			Most Human Human: Chapter 10,	PS5

		text editor	p.219-237	
11	2/15 Mon	optional and keyword parameters;		
12	2/17 Wed	Tuples; nested data structures and nested iteration	Most Human Human, Chapter 10, p. 237-259	PS6: Hangman game
13	2/22 Mon	Midterm exam review		
	2/23 Tues	Midterm exam 7PM		

	2/24
	Wed
		Lambda expressions;
		sorting;
		test.testEqual
				PS7: sorting (due 3/6, end of break)

14	3/7 Mon	TCP/IP; HTTP; URL lib; Unix curl; REST APIs; JSON lib; Unicode		

`15	3/9
	Wed
		String output formatting;
		CSV files;
		Try/except;
		FlickR API
			Success of Open Source: Preface and Chapter 1	PS8: FAA

	3/10 7PM Thurs	Second-chance midterm		
16	3/14 Mon	Defining classes		
17	3/16 Wed	Class inheritance	Success of Open Source: Chapter 2	PS9: classes basics; Flickr tag recommender
18	3/21 Mon	Oauth2; requests_oauth lib; FB lib; git		
19	3/23	More on classes: dealing cards	Success of Open Source:	PS10: FB emo scores

	Wed		Chapter 3	
20	3/28 Mon	Pyglet		
21	3/30 Wed	Test suites	Success of Open Source: Chapter 4	PS11: Pong game
22	4/4 Mon	List comprehensions	Success of Open Source: Chapter 5	
23	4/6 Wed	Twitter API	Success of Open Source: Chapter 6	PS12: test suites
	4/7 Section	Project plan reviews		
24	4/11 Mon	Comparative iteration pattern—the win streak estimator program; project coding time		
25	4/13 Wed	Madlib generator; Project coding time		
	4/17	Hackathon (coding party)		
26	4/18	Project showcase (selected projects)		Final projects version 2.0 due 4/18 at 2:30
	4/19 2:30-4PM	Review session for final exam		
Final Exam	4/20 1:30-3:30PM			