

ECE122 Introduction to Programming for ECE Syllabus

Instructor: Prof. Eric Polizzi
Office: Marcus 201C
Web: <https://people.umass.edu/polizzi/>

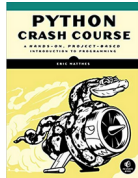
Day and Time: Lectures: M-W-F 11:15am-12:05am THOM 104
Discussions/Labs: Tuesday or Thursday (10-11:15; 1-2:15)- ELAB 304

Textbook (not required, some suggestions):

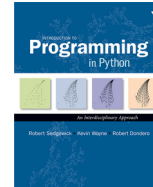


public domain:

<http://greenteapress.com/wp/think-python-2e/>



(clear/basic tutorial, HS basic level)



(college level with applications)



(college level, CS focus)

Resources on-line: main website is www.python.org

Outline of the Course

This course represents a **comprehensive** introduction to computer programming with applications to various areas in sciences and engineering. This course introduces the basic syntax of the Python programming language, the procedural programming paradigm, the object oriented programming paradigm, basic notions of scientific computing and the basic of data structures and algorithms.

Chapter 1: Getting Started with Python

1.1 Variables-Type 1.2 Input/Output (I/O)- Python Files- Programming mode

Chapter 2: Elements of Programming

2.1 Functions; 2.2 Comments- Conditional Statements (if/elif/else)
2.3 Notion of Algorithms – Modules; 2.4 Notion of Data-Structure – Lists
2.5 Iterations: for loops; 2.6 Iterations: while loops; 2.7 Data Objects
2.8 Data Objects Complement-Procedural Programming; 2.9 Application: Create quiz

Chapter 3: Object Oriented Programming with Applications

3.1 Method Objects; 3.2 Method Objects Complement; 3.3 More on I/O-Reading/Writing
3.4 More on I/O- Graphics; 3.5 Class Anatomy – Encapsulation;
3.6 Encapsulation and Properties; 3.7 Inheritance; 3.8 Polymorphism

Chapter 4: Scientific Computing in Python

4.1 Introduction; 4.2: Numpy arrays; 4.3 Random and Matplotlib

Chapter 5: Data Structure and Algorithms-

5.1 Tuple-dictionary-set; 5.2 Unsorted vs Sorted Lists-Binary Search Algorithm;
5.3 Simple Sorting: bubble, selection, insertion; 5.4 Stacks and Queues; 5.5 Recursion

Complement: Exceptions and discussion about objects in Python

Course Goals

At the end of this course, you should have learned:

- How to program (using a lot of programming practice)
- Python syntax and fundamentals
- Object oriented programming techniques
- How to solve engineering/scientific problems with programming
- Basic data structures and algorithms

Class Scheduling

		Discussions		Discussions	
	Mon	Tue	Wed	Thu	Fri
Feb	6 (lec intro)	7 (lab intro)	8 (lec 1.1)	9 (lab intro)	10 (lec 1.2) Q
	13 (lec 2.1)	14 (lab 1)	15 (lec 2.2)	16 (lab 1)	17 (lec 2.3) Q
	N/A	21 (lab 2)	22 (lec 2.4)	23 (lab 2)	24 (lec 2.5) Q
March	27 (lec 2.6)	28 (lab 3)	1 (lec 2.7)	2 (lab 3)	3 (lec 2.8) Q
	6 (lec 2.9)	7 (lab 4)	8 (lec review)	9 (lab 4)	10 (lec review)
	SPRING BREAK				
	20 (lec 3.1)	21 (lab 5)	22 (lec 3.2)	23 (lab 5)	24 (lec 3.3) Q
	27 (lec 3.4)	28 (lab 6)	29 (lec 3.5)	30 (lab 6)	31 (lec review)
April	3 (lec review)	4 (Mid-term)	5	6	7 (lec 3.6)
	10 (lec 3.7)	11 (lab 7)	12 (lec 3.8)	13 (lab 7)	14 (lec review) Q
	N/A	N/A	19 (lec 4.1)	20 N/A	21 (lec 4.2) Q
	24 (lec 4.3)	25 (lab 8)	26 (lec 4.4)	27 (lab 8)	28 (lec 4.5) Q
	1 (lec 5.1)	2 (lab 9)	3 (lec 5.2)	4 (lab 9)	5 (lec 5.3) Q
	8 (lec 5.4)	9 (lab 10)	10 (lec 5.5)	11 (lab 10)	12 (lec review) Q
	15 (lec review)	16	17		

New Project
Project due on Moodle at 11:59pm
Q new Quiz (from 5pm to 11am next Lecture)

Class Meetings

There are two types of class meetings held for this course:

- Lectures will be held three times a week. The main goal of the lectures is to present and discuss the main content that is covered in this course.
- Discussion/Lab activities are organized weekly. The goal of the discussion/lab sessions is to reemphasize the topics covered in the lecture and illustrate the concepts using specific code examples. Students will work on graded activities.

•**All class materials will be posted on Moodle**

These components of the course are designed to provide ample opportunity for you to clarify reading assignments, ask questions, and practice your skills. You are encouraged to seek any additional help you need during office hours.

Grading

Your final grade will be calculated from your performance in four areas:

- 10 Quiz: quick assignment throughout the semester to keep up with the course materials. Check the schedule.
- 10 Lab Activities: Google colab (python notebook) activities.
- 4 Projects: Those are larger and **time consuming** coding assignments that require the development of solutions to practical problems.
- 2 Exams which consist of one mid-term exam and one final exam. The goal of the exams is to evaluate how well you retained and understood the course content as well as how well you can apply the course concepts to new problems. In-class review session will be held before each exam.

Moodle Quiz and Projects are assigned according to the schedule posted on the course website. Solutions to project assignments must be submitted on Moodle. Midterm is held according to the schedule on the course Moodle website. The final exam is scheduled by the university.

The final grade will be norm-referenced (i.e., **graded “on a curve”**) with the following weights:

- Quizzes: 8%
- Lab Activities: 20%
- Projects: 32%
- Exam: 40% with the following formula $(1/3)*\min(\text{midterm}, \text{final}) + (2/3)*\max(\text{midterm}, \text{final})$

Make sure to track your scores on Moodle to ensure that you have received the appropriate credit for each of your assignments and exams. No “make-up” assignments will be given (with exception to the cases stated in the examination policy below).

Some Course Policies

Class start at 11:15am. Late arrival is distracting and inconsiderate.

Exams will be based heavily on class notes, projects and discussions. Examinations assume familiarity with all lectures, projects and lab activities. You are expected to attend all classes and are responsible for knowing the material and assignments from every class.

Scheduling conflicts regarding exams should be reported to the instructor **immediately**. In case of a medical emergency, make-up exams will be given only if you provide a valid written excuse (as defined in Undergraduate Rights and Responsibilities) and notify me prior to the missed exam. Other missed exams will be considered failures. If advanced notice is possible and not given, the instructor may refuse the request.

Due to the very large class size technical emails should be sent to TAs, email-based requests for project assignment will be limited to a maximum of three per assignment per student. If you need further help, we highly encourage you to make use of office hours. If you are having difficulties with any of the material (or basic background), come to office hours. Do so before you fall behind.

Short general questions may be posted on **moodle forum** for all the students to see (**no sharing of code**). Other students, TAs or instructor will provide a response.

Project Assignments are due as posted on the course Moodle web page. **Late submissions will never be accepted.**

For Projects, consultation with fellow students is encouraged. However, **directly copying another student's work (past or present) defeats the purpose of the assignments and exams and is an honor code violation.** Unless otherwise noted, you are expected to complete all assignment individually. Violations will result in serious penalties including course failure and possible disciplinary action. If in doubt, please consult the instructor or the official UMass guidelines regarding academic honesty.

Missing quizzes or missing lab activities

There will be **no make-up**. You are allowed to miss 2 quiz and 2 lab assignment without the need to provide any excuses, you will get a 0 grade but the lowest grade will be dropped at the end of the semester. If you miss a third time, however, you need to provide valid excuses (doctor notes) for at least one of the dates that you have missed. If you miss a particular lab activity because of sickness or some good reasons, you are welcome to attend any other lab sessions (if space permitted).

Missing projects

Project extension beyond the deadline is not allowed (no exception, no extension policy). If you have a valid excuse (doctor notes), you need to upload it on Moodle before submission.

Warning: you have two weeks for completing your projects but you are supposed to get started right away. If you have a valid excuse such as you have been sick for one week (with doctor note), you are still expected to have worked one week (~50% of project completion) and your unfinished project must then be submitted on-time.

Warning: Excuses that are **not-valid**

Excuse 1: The server was down – *not likely*

Excuse 2: I “accidentally” uploaded the wrong files or my zip file is empty– *do not forget to check your file immediately after submission.*

Excuse 3: My network connection went down – *make sure assignment is uploaded early (you can keep uploading new versions until deadline)*

Excuse 4: I “accidentally” erased all my Files- *it could happen (happened to me a few times...), life lesson, create extra copies (cloud storage, etc.)*

Excuse 5: I was stuck on a deserted island-....Tom Hanks, is that you?