Object-Oriented Programming
50:198:113 (Spring 2019)

Handout:	**1**	**Professor:**	**Suneeta Ramaswami**
Date:	**1/23/19**	**E-mail:**	`suneeta.ramaswami@rutgers.edu`
Office:	**321 BSB**	**URL:**	`http://crab.rutgers.edu/~rsuneeta`
		Phone:	**(856)-225-6439**

Course Outline

Course Description:

This course builds upon the elementary programming skills learned in the Programming Fundamentals (50:198:111) course and involves a deeper study of the principles of object-oriented program design and algorithmic problem solving. We will program in Python although most of the material can also be adapted for designing and implementing programs in other similar object-oriented languages such as C++ or Java.

If you are a student that has transferred equivalent credits for 198:111, but do not have a prior background in Python, you are expected to catch up with Python basics (Chapters 1 to 17 of the textbook) in the first two weeks of the semester. I will review some of this material in class over the first two to three weeks (look at the *Schedule of Topics* below), but please keep in mind that at the end of that period (say by February 7), you should have an adequate knowledge of Python programming fundamentals. You might find the chapter quizzes a good way to test yourself on the fundamental topics and to identify potential topics of weakness early in the semester.

Course Material:

All course-related material such as the syllabus, programs covered during lectures, homework assignments, solutions to homeworks, and any announcements, will be available on the Sakai site for this course. You are expected to check the course Sakai site regularly.

Text Book:

Required: *Learning Python* (5th edition), by Mark Lutz, O'Reilly and Associates. ISBN 978-1-449-35573-9.

Office hours:

Tuesdays 10:30AM - 12:30PM, or by appointment at a mutually agreeable time.

Course Work:

1. Six Programming Assignments, worth a total of 400 points (#1 and #2 are worth 50 points each; #3, #4, #5, and #6, are worth 75 points each).
2. One in-class exam, worth 250 points.
3. Final, worth 350 points.

Tentative dates for Assignments and Exams:

HW #1:	Out 1/28, Due 2/11	**HW #2**:	Out 2/11, Due 2/25
HW #3:	Out 2/25, Due 3/8	**HW #4**:	Out 3/8, Due 3/25
HW #5:	Out 3/25, Due 4/15	**HW #6**:	Out 4/15, Due 5/6
Midterm:	Wednesday, 3/13	**Final**:	Monday, 5/15, 8:00-11:00am

Homework and Collaboration Policy: As the second course in the sequence of *programming* courses for CS majors, this course is designed to be a hands-on course in which you spend a substantial amount of time writing and debugging programs. The depth and amount of material covered is substantial. Just reading the textbook will not be sufficient to understand the material. **Come to every lecture!** The lectures will be interactive: we will write and run programs in class, and you will be expected to ask, and answer, questions.

It is especially important that you start your assignment early enough in order to complete it by the deadline. I fully expect you to need the time you are given (2 to 3 weeks) to complete each assignment, so do not start working on your homework assignments at the last minute!! If you do, chances are high that your programs will not work correctly, or at all. Homework assignments **must** be submitted on the due date. Late hand-ins will be subject to a 50% deduction per day after the due date. There will be no incomplete grades handed out for this course (unless there is a compelling reason such as a certified health problem).

General discussion of course material with fellow students is allowed and encouraged. However, *all work on the homework assignments must be done independently.* Please respect this policy. Violations will be dealt with harshly.

Tentative Schedule of Topics to be Covered

To get the maximum benefit from the lectures, it is strongly recommended that you make an initial reading of the chapter(s) indicated for each week *prior* to the lecture.

Dates	Topics	Reading
1/23, 1/28	Review of Python Fundamentals	Chs. 1-15
1/30, 2/4	Functions and Scopes	Chs. 16-17
2/6, 2/11, 2/13	Advanced Function Topics	Chs. 18-20
2/18	Modules	Chs. 22-23
2/20, 2/25	Object-oriented Programming	Chs. 26-27
2/27, 3/4	Classes and OO design	Ch. 28
3/6, 3/11	Implementing classes	Ch. 28-29
3/13	**In-class Midterm Exam**	All material covered upto 3/6
3/18, 3/20	*Spring break*	
3/25, 3/27	Implementing classes (continued)	Ch. 29
4/1, 4/3	Polymorphism through operator overloading	Ch. 30
4/8, 4/10, 4/15	Polymorphism through inheritance	Ch. 31
4/17, 4/22	Exception handling in Python	Ch. 33-34
4/24, 4/29, 5/1, 5/6	Elementary data structures	class notes