COMP 150
Introduction to Computing
Spring, 2022

Course #:	Section 002, asynchronous 2162
	Section 004, synchronous 3488
Day/Time:	Tuesday, 4:15 – 6:45 online
Zoom Room:	https://luc.zoom.us/my/cnaiman
Prerequisites:	None
Instructor:	Dr. Channah Naiman
Email:	cnaiman@luc.edu
	Be sure to put your course name in the subject line!!
MS Teams Course Code:	fdwdglk

Orientation Video

Description (from catalog)

This course provides a broad survey introducing the many layers of the computer science discipline, emphasizing the computer's role and limitations as a tool for describing, organizing, and manipulating information applicable to many disciplines. Topics include binary logic expressed in electronic circuitry, machine architecture, basic programming in the very accessible language Python, data organization, the potential and limitations of machines, and useful tools.

This course serves as a terminal course for students who want a one-course introduction to the field, as well as a preliminary course to upper-level computer science offerings.

Table of Contents

- Prerequisites
- Textbooks and other course materials
- Tutoring
- Course Objectives and Goals
- Software
- Course Format
- Exams
- Academic Honesty
- Lateness Policy
- Course Components and Grading
- Course Schedule
- Important Dates
- Professor Notes

Prerequisites
None

Course Objectives and Goals

- To introduce the concepts of algorithms and their analysis and design in the context of a simple yet powerful computer language (Python)
- To design, test, and debug Python programs to implement various fundamental programming constructs such as basic computation, simple I/O, data structures, standard conditional and iterative structures, and functions,
- To learn how to add graphical effects to programs.
- To understand the building blocks of hardware components and multi-layer language constructs, from machine instructions to low level and assembly languages to high level languages like Python.
- To understand the building blocks of HTML and construct a web page containing forms and write a Python CGI script to dynamically process information placed in a web form (if time permitted)

Textbooks (and other course materials)

There is no specific text for this course. We will be using resource materials available online.

- *For the Python component of the course:*
  - We will be using The Hands-on Python Tutorials designed specifically for this course by Professor Andrew Harrington. It is available on the web, both in browsable format and in printable pdf format. Professor Harrington also created detailed videos on One Drive and also on Google Drive to walk you through the tutorials. This is will serve as our "textbook" for the course. Please see below for other free online references for Python. The tutorial was originally written for Python 3.1, and it has been updated many times since then. For the purposes of this course, any version of Python 3.x will work. The basic constructs of the language are the same for all 3.x versions.
  - Example files are referenced in the tutorials, and you may download there here.
  - Strongly recommended: https://pythonprogramming.net/else-python-3-tutorial/
    This has nice short video examples of many programming topics.
- *For the non-Python topics*, there are separate videos and short labs. Class notes for these topics are linked to from the Course Schedule at the end of this syllabus.

Free Online References for Python:

There are thousands of reference materials for python. Below is a selected list of python tutorials and examples that you may find helpful. All except the official Python Tutorial are intended for people with no programming background. Some of the links still have versions for Python 2.x, which has some important syntactic differences from Python 3.x, so look for the version support. Support finally ended for Python 2.x on January 1, 2020, so hopefully you won't run into it too much. For the purposes of this course, pretty much any version of Python 3.x will be fine.

- http://python.org is the home site for the Python language, where you can find out more and download the latest production version starting with a 3 (3.9.5 in June, 2021).
- An interactive web version of How to Think Like a Computer Scientist starts with pretty much the same basics as the Hands-on Python Tutorial, but in a somewhat different order with a different graphics library. It goes much further, allows you to program totally inside the browser, and has embedded videos. There is also an older, somewhat different, non-interactive pdf version if you prefer to work offline.
- http://www.pythontutor.com/ is a wonderful site where you run code in your browser step by step and can save and email a URL reference to any step. It supports up to Python 3.6, but that shouldn't make a difference for this course.
- https://medium.com/swlh/5-free-python-courses-for-beginners-to-learn-online-e1ca90687caf As the link states, this site has links to many free courses for python.
- https://www.youtube.com/watch?v=_uQrJ0TkZlc Another good video tutorial, with a useful table of contents.
- http://en.wikibooks.org/wiki/Non-Programmer's_Tutorial_for_Python_3 is another introduction to Python basics in different words. No graphics, few major examples, lots of basic syntax.
- Extra Material on IDLE errors messages – this may prove very useful in the future!
- Official Python 3 Tutorial *moves very fast for someone with no programming background*. It covers a lot,

going well past our course! This will be a great resources to keep for future reference once you've become Python masters!
- [Software Carpentry - Python](#) is an intro for data scientists using the numpy library and the nice development/display environment ipython.
- If you find none of these helpful, there is a much longer list at http://wiki.python.org/moin/BeginnersGuide/Programmers

Tutoring Assistance (free!):
- Tutoring hours by Computer Science TA's (generally first-come-first-served) is posted at http://www.luc.edu/cs/academics/tutoring/. It should start by the second or third week of the semester. The tutors are a great way to get more individualized help for specific needs. If any tutors you visit are NOT familiar with Python, let me know right away!

- The university supports an official Tutoring Center in the Sullivan Center. Their phone extension is 8-7708. Their web site is http://www.luc.edu/tutoring. You may need to make appointments in advance for university tutoring. When consulting the university tutoring center hours, remember that, at most, tutors for Comp 150 are only available during some of those times, due to the personal schedule of the tutor(s). We can't guarantee that they will have tutors knowledgeable in Python. In general, COMP students have done better with the department's tutoring service.

Software
- *Python (and IDLE):* We will be programming in Python, available in the university labs, and as a download to your own computer. There are several choices based on program version and operating system. Current version of Python is 3.9.5 (as of June, 2021), but any 3.6 or greater version will work. (Actually, pretty much any 3.x version will work for this course.) Download the latest version for your operating system from the central site http://www.python.org/download It comes with the graphical interface, Idle, which we will use extensively.
- *Alternative environments:* There are many alternate free Python development environments, like [PyCharm,](#) Community Edition. If you are familiar with Jupyter Notebooks (or iPython), feel free to install Anaconda and use that. You could use Spyder instead of IDLE or PyCharm. We may demonstrate the use of Jupyter notebooks at some point in the class, time permitting. But again, PyCharm, Anaconda, Jupyter notebooks, Spyder, etc., all have learning curves of their own. Therefore, an inexperienced programmer may simply want to use IDLE, which is easy to use and comes already installed with Python. However, you can complete the assignments in any IDE of your choice. I will be using IDLE to demonstrate programs in this course, and the tutorial videos also use IDLE.
- *Storage media & Backup:* If you are working on your own laptop, then save your assignments and projects in a folder that you can easily find later. If you are working on a LUC public computer, be sure to save your work on a flash drive, on Google Drive, on One Drive, or back up on any server or location of your choice. As with all public computers on campus, in the event of a logout, timeout, or a power outage, all temporary files will be lost.
- *Pre-installation notes:* Before you install anything, be sure to look at the [extra Windows notes](#) or [extra Mac notes.](#)

Course Format: This course is scheduled as an online, flipped course. One section is synchronous, and one section is asynchronous. However, all students are welcome to attend the Tuesday night zoom sessions, which *may* also be recorded (no promises). These terms are defined below:
- *Online:* This means that we will not be meeting in person, on campus. The course is fully online. Lectures ("flipped", see below), Q & A, office hours, student and team meetings with me, exams, and project presentations will all be conducted online, either in my zoom room or, for exams, on Sakai.
- *Synchronous:* This means that Section 004 will have zoom meetings on Tuesdays, from 4:15 – 6:45 p.m.
- *Asynchronous:* This means that Section 002 has no scheduled meeting time. However, all students are welcome to join the Tuesday sessions.
- *Flipped:* This means that you are responsible to watch the tutorial videos and follow along with the tutorial examples before coming to class, at your own speed, and come prepared to our class sessions with questions. This allows us to take better advantage of our class time to review difficult examples, and get clarification of the more challenging concepts and assignments. During the first two sessions, I will most likely lecture (depending on how prepared students are for those sessions), and I may lecture and review for *part* of the other synchronous sessions, depending on the difficulty of the material, or the questions that I get from students. Hopefully, this approach will give you the most bang for the buck in using all of the resources that this course makes available to you. **Of course, you may *always* ask me questions**

outside of class time, either by email, MS Teams, or individual/group zoom meetings.

- *Important notes about this format:* The success of any online course, and especially one using a "flipped" format, depends on your commitment to come prepared to the synchronous sessions. If you come prepared, we can spend the time reviewing programming problems that you have found challenging, and discussing concepts that can help you with your homework and project. If students do not come prepared, then we may spend the time simply going over the programming examples in the tutorials. I'm hoping for the former, but it's going to be up to you.

Exams: There are two exams. Tentative dates are:

- Exam #1: 2/15
- Exam #2: 4/05

We will have a final Project instead of a final exam. You may opt for a video presentation. If you opt for a zoom presentation, then the final exam time slot will be used for Project Presentations. If there are too many presentations for that time slot, we'll have additional time slots earlier.

Exams will cover material discussed in class, reading material on the web, and assignments. Exams will always be cumulative. They will be administered on Sakai during regular class time, and will be either 1.5 hours or 2 hours in length. We may have class following the exam. Because the exams are online, I cannot enforce a closed books or closed computer policy. However, the exams are timed, so it is important to be prepared. I would strongly suggest that you prepare t two 8.5 x 11 inch sides of notes for exams, with examples of some of the important programming constructs that we learn. I emphasize having you process and use information, not regurgitate facts -- put the facts you most forget and still need in your notes. This is very different than many of the requirements some students had in high school, where fact recall may have been key. What you want most to remember is general patterns about the process of breaking problems into pieces and identifying the right process for each piece. The pieces may come in all sorts of combinations, so remembering whole rote sequences is not likely to be helpful. Do not depend on it.

There is a review for each exam posted on the Course Schedule.

If you must miss an exam, let me know well in advance. Then if you have an acceptable and verifiable reason we can possibly make other arrangements. Exam absences will be verified. If they can't be verified, you will not be excused. If an emergency arises on the day of the exam, please let me know as soon as possible, and provide verification. If approved, I will determine the nature of the accommodation.

No second try: If you have an excuse for not being prepared to take an exam, but decide to take it anyway, you don't get to change your mind after you see a poor grade. Being sick is not a way to get one more chance than everyone else. I may allow you to delay an exam due to illness, but I will not let you be reexamined due to a poor grade.

Academic Honesty

Students are expected to have read the statement on academic integrity available http://www.luc.edu/academics/catalog/undergrad/reg_academicintegrity.shtml This policy applies to the course. The *minimum* penalty for academic dishonesty is a grade of F for that assignment. Multiple instances or a single severe instance on a major exam or assignment may result in a grade of F for the course. All cases of academic dishonesty will be reported to the department office and the relevant college office where they will be placed in your school record. Academic dishonesty includes, but is not limited to, working together on assignments that are not group assignments, copying or sharing assignments or exam information with other students except in group assignments, submitting as your own information from current or former students of this course, copying information from anywhere on the web and submitting it as your own work, and submitting anything as your own work which you have not personally created for this course. If you do wish to use materials that are not your own, please check with me ahead of time and cite you source clearly. When in doubt, ask first!

Please remember that you must be able to explain your code. Please think very carefully about submitting sophisticated code that is far above what we are covering in this course. If you do use it, make sure you understand it, or it will be considered plagiarism.

Lateness Policy

Due dates. Assignments are due as specified in the syllabus Course Schedule and on Sakai. Some are on Tuesdays, at 11:55 p.m.. Some, such as some orientation assignments, must be submitted before a specific class session begins, so that you will be prepared to participate in that day's lab. For some, I give you one or two extra days after class, and for some, you will have over the weekend to complete the work. All assignments and due dates are posted on the Course Schedule on the Syllabus, and also on the Course Calendar on Sakai.

Late Credit. Do not assume that there is an automatic "half credit" for late assignments. There is not.

Extensions and "submit until". Any extensions in due dates will be sent as email announcements on Sakai. In the rare event that I allow an individual student to submit an assignment late, it will be graded as half credit. **Some assignments have a "submit until" date listed on the Sakai assignment. That is not the due date. The "submit until" date is only valid when I give permission to a student or to the class to extend the due date.**

Help at the last minute. The purpose of the due dates is so that you won't fall behind. I take them seriously. So should you. It is in your best interest NOT to wait until the last minute to begin working on your homework and labs. I cannot guarantee that I will be able to help you on the due date. I have many other students, and generally, when a student waits until the last minute, he or she is less prepared and needs even more time.

There's no such thing as an emergency. This is an online class. Assume that there will be technical issues, or that your internet connection may occasionally go down. Barring a catastrophic internet disaster, or a true last-minute medical emergency, there is no such thing as an emergency. There is only poor planning.

Online Recording Policy

Many lectures and labs are prerecorded. However, often students ask me if I can also record the online sessions. If we do record the online sessions, I will announce that I am about to record, and students can opt to turn off their video. Any recordings will comply with the official Loyola Recording Policy:

Loyola official recording policy: In this class software will be used to record live class discussions. As a student in this class, your participation in live class discussions will be recorded. These recordings will be made available only to students enrolled in the class, to assist those who cannot attend the live session or to serve as a resource for those who would like to review content that was presented. All recordings will become unavailable to students in the class when the Sakai course is unpublished (i.e. shortly after the course ends, per the Sakai administrative schedule: https://www.luc.edu/itrs/sakai/sakaiadministrativeschedule/). Students who prefer to participate via audio only will be allowed to disable their video camera so only audio will be captured. Please discuss this option with your professor. The use of all video recordings will be in keeping with the University Privacy Statement shown **_below_**:

Privacy Statement
Assuring privacy among faculty and students engaged in online and face-to-face instructional activities helps promote open and robust conversations and mitigates concerns that comments made within the context of the class will be shared beyond the classroom. As such, recordings of instructional activities occurring in online or face-to-face classes may be used solely for internal class purposes by the faculty member and students registered for the course, and only during the period in which the course is offered. Students will be informed of such recordings by a statement in the syllabus for the course in which they will be recorded. Instructors who wish to make subsequent use of recordings that include student activity may do so <u>only</u> with informed written consent of the students involved or if all student activity is removed from the recording. Recordings including student activity that have been initiated by the instructor may be retained by the instructor only for individual use.

Religious Holidays

Please let me know within the first two weeks of class if you have a religious holiday conflict with any exam or homework due date so that we can plan on a makeup schedule.

Students with Disabilities

Loyola's Official Policy: Loyola University Chicago provides reasonable accommodations for students with disabilities. Any student requesting accommodations related to a disability or other condition is required to register with the Student Accessibility Center (SAC). Professors will receive an accommodation notification from SAC, preferably within the first two weeks of class. Students are encouraged to meet with their professor individually in order to discuss their accommodations. All information will remain confidential. Please note that in this class, software may be used to audio record class lectures in order to provide equal access to students with disabilities. Students approved for this accommodation use recordings for their personal study only and recordings may not be shared with other people or used in any way against the faculty member, other lecturers, or students whose classroom comments are recorded as part of the class activity. Recordings are deleted at the end of the semester. For more information about registering with SAC or questions about accommodations, please contact SAC at 773-508-3700 or SAC@luc.edu.

Additional notes for this course:
- I am not allowed by SAC to provide accommodations retroactively. Please apply for your accommodations in time for the semester.
- I am also not allowed to accommodate you beyond what is specified in your accommodations letter. This is in the interest of being fair to all students.
- Please do not send me any personal medical information. If you have a chronic condition that requires accommodation, you MUST go through SAC. I cannot accommodate you without a letter from SAC. If you have an emergency situation that truly could not have been anticipated earlier, please do not share the details with me if they are of a personal nature. You may speak with your advisor or the Dean of Students, who can then contact me.

Students with Sponsorships and Scholarships

If you require a certain grade in order to satisfy a sponsor or a scholarship requirement, please be sure to monitor your grade on Sakai. I will consider only your performance in this course in calculating grades, using the grading rubric posted in this syllabus. If you cannot achieve a minimum grade that is required by a sponsor or a scholarship, I will not change your grade to help you meet that requirement. This would be unfair to other students, and not reflective of your performance in this course. You are responsible to monitor your grade and to keep apprised of the withdrawal dates posted on the LUC Academic Calendar.

Course Components and Grading

Orientation Activities	25
Tutorial, Chapter 1 Exercises	55
Tutorial, Chapter 2 Exercises	60
Tutorial, Chapter 3 Exercises	70
Exam #1	200
Exam #2	200
Python Project	200
Python Project "Excellence" points	50
Pip Program	60
Gates Homework	60
Participation	50
Total	**1030**

Extra Credit points:

Tutorial, Chapter 2 Exercises:	
Texttriangle.py (Exercise 2.4.12.2)	3
Tutorial, Chapter 3 Exercises:	
chooseButton4.py (Exercise 3.2.1.1, part d)	3
makePath.py (Exercise 3.3.4.2)	6
madLib4.py (Exercise 3.3.4.4)	5
Homework Chapter 4	15
Total	**32**

- *Grade Calculation:* Note that there are 1030 total possible points in the regular "required" course components. *The course is graded on the basis of 1000 points.* Thirty extra points are built into the course.
- *Understanding your work:* All submissions are subject to your ability to explain your work to me. If you submit perfect programs or exam questions, but you are unable to explain how you wrote/ran the program or how you arrived at a answer, you will not get credit for that part of the assignment.
- *Extra Credit:* In addition to the 30 extra points built into the regular, required course components, there are 32 extra credit points available in the tutorial homework assignments. No further extra credit opportunities will be provided. It is neither practical nor fair to the other students.
- *Project "Excellence" Points:* Python Project "Excellence" points are *not* extra credit. A "very good" project that fulfills the requirements and is perfectly satisfactory can earn full points for all non-Excellence (200) points for the Project. If you get full points on your project and do not earn any Project Excellence points, please don't ask me what you did "wrong". The answer will most likely be "nothing". Project Excellence points can be earned for a project that is truly excellent, incorporating components well beyond the material covered in class, or simply having a real "Wow" factor. A grade of "A" means "excellent". Full points for Homework and Labs can be earned if significant effort is demonstrated: you are graded for completeness, not necessarily for correctness (although grossly incorrect answers will be marked off). But for a project to get the full 250 points, it must be excellent. As I review your progress during project walkthroughs, I will tell you what you can do to earn excellence points for your project. I expect that every team that puts in full effort into the project will earn at least some (and probably most) project excellence points.
- *Participation:* Your participation grade is based on your attendance at our synchronous meetings (for the synchronous session, unless I give you a pass); participation at project team meetings with me; participation in your team project; questions, emails, MS Teams postings; timely submission of assignments.
- No further accommodations will be provided. If your personal circumstances are such that you have to miss more than 50 points worth of material, please consider taking the course another semester.
- *Team Participation and grading:* Do not assume that all team members will automatically earn the same grade on the project. Your participation and contribution will be assessed by a combination of your ability to answer questions during the zoom team project "walkarounds", the team participation

form, and your knowledge of the project and code before, during and after the project presentation.

Course grades are assigned as follows:

A	93
A-	90
B+	87
B	83
B-	80
C+	77
C	73
C-	70
D+	67
D	63

Course Schedule
Please note that the official Academic Calendar, found here: www.luc.edu/academics/schedules, has information about all Loyola breaks, the final exam schedules, and other schedule information.

The dates below give the sequence and a general idea of the time spent, though we may get ahead or behind this time schedule at different points, depending on the needs of the class.

Prep: means preparation done *before* class, with your questions always noted for discussion in class! Sometimes extra **In class** activities (besides questions and active use of preparatory reading) are mentioned. Written assignments should be turned in to Sakai by the end of the day listed (11:55PM), unless otherwise noted. The section on Course Materials discusses how to obtain the videos for the listed sections.

Week	Date	Activities	Assignments
		Before class **Orientation:** • Course Introduction and syllabus screenshot • Python Installation • Introduce Hands-on Python Tutorial , Example Files • Greetings forum (on Sakai) • MS Teams signup	Orientation assignments are posted on Sakai. They are due on 1/24 at 11:55 p.m. Earlier is better!
1	1/18	**Prep**: Hands-on Python Tutorial through Input/Output 1.10 (most video for one class, with all the introductory material); Much of this has been completed during orientation	
2	1/25	Catch up and questions: 1.1 - 1.10 **Prep**: Python Tutorial functions 1.11, dictionaries 1.12	
3	2/01	**Prep**: Python Tutorial 1.13 Loops	
		Prep: Python Tutorial 1.14, hw questions ready!	
		In class: Review materials for Exam #1	
4	2/08	**Prep**: Tutorial 2.1 - 2.2 (2.3 optional but strongly recommended)	
		Prep: Tutorial 2.4 , 2.4.1, 2.4.2	
5	2/15	Questions, examples on 2.4.1 and 2.4.2, time permitting Exam 1 (review materials) • Part 1: Change existing code to match output • Part 2: Write the code	Chapter 1 Exercises Due 2/15 at 4:00 p.m.
6	2/22	**Prep:** Tutorial 2.4.5-8, Color (optional, recommended): 2.4.10 – 2.4.12	
		Prep: Tutorial 2.5	
7	3/01	Discuss Python Project (finalize teams) **Prep**: Python Tutorial Chapter 3 through 3.1.6	Chapter 2 Exercises DUE on 3/05 at 11:55 p.m. Accept until 3/10 at 11:55 p.m.
8	3/08	**Spring Break**	
9	3/15	**Prep**: Python Tutorial 3.1.7 - 3.3.1	

10	3/22	**Prep**: Python Tutorial 3.3.2. - 3.3.4 through bounceWhile.py **Project Discussion**	
11	3/29	Chapter 3 Review and Exam Review Questions on Homework 3, part 2	Chapter 3.1 Exercises Due 3/29 at 11:55 p.m.
		Prep: Bases and Binary Arithmetic in binary topic notes (videos N2, N3A, N3B, N3C-E)	
12	4/05	Exam 2 (review materials)	Chapter 3.2--3.3.4 Exercises Due 4/05 at 4:00 p.m. before class
13	4/12	**Prep**: Pip assembler through 4E in topic notes (videos N4, N4A, N4B, N4C, N4D, N4E)	Submit any independent plan for the Python Project
		Prep: Pip If-else in assembler (Videos N4F, N4G, N4H)	
14	4/19	**Prep**: gates and Boolean algebra (N5A-D) **Prep**: Gates and Boolean Algebra (N5E-I) *Please note that we will use a different gates simulator than the one shown in Dr. Harringtons's videos. This will be demonstrated in class and/or a video.*	Due 4/19: Pip Program Due 4/23: Gates HW
15	4/26	Project Work and/or catch up Possible Project Presentations Possible team zoom meetings instead of regular class	Due 4/23: Gates HW (Extra credit homework: Chapter 4 Exercises)
16	5/03	Project Presentations: video or zoom at regular class time as per LUC final exam schedule	DUE 5/03 at 6:45 p.m.: Python Project Due Submit: - Files - PPTs - Team Partiipation form

Important Dates: Keep track of the withdrawal deadlines on the **LUC academic calendar:**