TEXAS
The University of Texas at Austin

I304: Programming for Informatics

Spring 2025

Overview

This class offers an introduction to computer programming for those **without any prior knowledge or experience in computer programming**. If you already have programming experience, please explore options below for waiving the requirement to take this class

Requirement to complete I304

This class is required for the undergraduate B.S. Informatics degree (see concentrations) and as a prerequisite for various other informatics courses. Alternative ways to meet this requirement include:

- Complete this class (see Past & Future Offerings of this Course)
- Complete a suitable substitute course
    - BME 303 Introduction to Computing
    - CS 303E Elements of Computers and Programming
    - CS 312 Introduction to Programming
    - J 326C Introduction to Coding for Journalists
    - LIN 313 Language and Computers
    - LA 319 Computer Programming: Liberal Arts
    - MIS/BAX 304: Introduction to Problem Solving and Programming
    - SDS 322 Introduction to Scientific Programming
- Transfer credit, e.g., from AP Test
    - https://utdirect.utexas.edu/ctl/cbe/petition/index.WBX
- Test out of the requirement
    - Exam in Computer Science 303E
    - Exam in Computer Science 312

Why Take this Class? (beyond the requirement above)

- Acquire valuable skills to apply in a job, during school or after graduation (whether or not you plan on this today).
- To better understand how computers work, since they are increasingly entwined in our daily lives and crucial to how our modern world functions.
- To better engage with AI, since AI is a powerful tool but only if you guide it.
- To create and be creative. Programming is artistic expression.

TEXAS
The University of Texas at Austin

Course Details

Meeting Days/Times: T/Th 12:30 pm - 2 pm
Unique ID: 28120
Location/Format: McCombs School of Business (CBA) 4.348 (in-person)
Instructor: Dr. Steph Buongiorno
Email: steph.buon@utexas.edu
Office Hours: T/Th 11 am - 12 pm

Note: Assume that the instructor will respond to messages during business hours, Monday - Friday. Please do not expect responses in the evenings and over the weekends; plan ahead for deadlines!

Catalog Description

Examine the fundamentals of computer programming and practice developing and documenting code.

Course Objectives

At the end of this course, students should:

1. Have the skills required to solve problems by creating and modifying programs in Python.
2. Have the knowledge of basic programming concepts (e.g. computational thinking, architecture, flowcharts, documentation, debugging, version control), their appropriate usage, and how and where to learn more.
3. Have the skills to communicate about programs and the programming process to help clarify their ideas and to communicate these ideas to others.
4. Have an attitude of confidence when reading, writing, or discussing computer code

Prerequisites

There are no prerequisites for this class.

Required Materials and Devices

Software

Python Development assignments will be completed using Google Colab, an approved browser-based development environment that allows students to write and run fully functional Python programs online. This eliminates the need to install Python on personal computers, ensuring all students use a consistent and suitable development platform.

Additionally, students will need access to a diagram drawing tool such as Draw.io, Miro, or LucidChart. We will default to using Draw.io because it is free and accessible online. However, students with prior experience using other diagramming tools are welcome to use those instead.

At the end of the semester, additional resources on installing Python locally and exploring other development environments commonly used in different areas of Python development will be provided to support students' continued learning. However, local installations are not required or used in this class.

Primary Textbook

We'll be using a new version of Charles Severance's excellent *Python for Everybody* (abbreviated as PY4E). The book is free and open. It is accessible as a PDF, a book, or a website online. Alternatively, you can purchase a hard copy for less than $15 online. **We will default to using the online version to complete exercises.** But there are a lot of ways to consume material. I wanted to provide these other options in case you have a preference (e.g. digital vs. hard copy). All of the online chapters can be accessed here. In the syllabus schedule I will include links to the relevant chapter(s) for that week.

Required Supplementary Materials

As a supplement to Charles Severance's *Python for Everybody* (abbreviated as PY4E), I have curated a series of Notebooks to further explain and reinforce programming concepts. These are accessible through Google Colab and are also required reading. In the syllabus schedule I will include links to the relevant Notebook(s) for that week.

Hardware

Students will need laptops able to run Firefox or other modern web browsers. iPads are NOT recommended for coding. If you must use an iPad, I highly recommend purchasing an external keyboard. Or, since iPad accessories are already pretty expensive, purchase a $150 Chromebook. Programmers need keyboards. University-owned machines may be available for the times you will need to record your screen if your machine is incapable of doing so. Contact me early if you'd like help securing access to hardware you don't have or can't afford.

Assessment

Course Grades (there is no rounding)

- 94 or above (A: superior), 90-94 (A-: distinguished)
- 87-90 (B+: good), 84-87 (B: satisfactory), 80-84 (B-: barely satisfactory)
- 77-79 (C+: unsatisfactory), 74-77 (C: unsatisfactory), 70-74 (C-: unsatisfactory)

For a course to count towards degree requirements, the minimum grade is a C-

Grades will be posted on Canvas as work is submitted over the course of the semester. **Feedback will be provided directly on the Google Colab notebooks**. Homework will be evaluated holistically, i.e., on the basis of overall quality.

Category	Percentage of Final Grade
Readings	10%
Canvas Assignments	60%
Final Project	15%
Presentation	5%
Participation	10%
Attendance	0%*
Total	**100%**

* *Attendance is mandatory, so unexcused, non-emergency absences will negatively impact a student's grade. (see below)*

Readings

Students will learn the fundamental concepts of programming as implemented in Python using: (a) Charles Severance's *Python for Everybody,* which is free online; and (b) supplementary Google Colab notebooks that explain and reinforce core concepts. Links to the supplementary notebooks are in the syllabus. **To receive credit for your reading, you must answer any questions at the end (if provided) and submit your answers on Canvas.** Questions are due 11:00 AM (before class begins).

Canvas Assignments

Canvas assignments may consist of two components: (a) Python development tasks and (b) diagram creation. Python coding should be completed using Google Colab, while diagrams should be created using a diagram drawing tool. Suggestions for diagram drawing tools are listed under "Software." Assignments are posted on Canvas every Thursday and are due the following Tuesday by 11:00 AM (before class begins).

Code. Python development tasks focus on building practical programming skills and reinforcing foundational concepts. These exercises aim to develop problem-solving abilities, coding proficiency, and familiarity with Python. Submitted code should include sufficient comments to explain key sections and ensure readability. Assignments will be evaluated based on function

and adherence to best practices (as expressed by the course up to the point in which the assignment was given). These best practices might include clear variable naming and logical organization.

While solutions are expected to function correctly, students will not be penalized for minor inefficiencies in implementation. The focus will be on demonstrating an understanding of the underlying principles and problem-solving process.

Diagrams. Diagrams, such as flowcharts and system diagrams, help students understand program structure and logic, bridging the gap between conceptual problem-solving and writing code. These exercises foster computational thinking and reinforce foundational programming principles. Additionally, diagramming is an invaluable tool for planning complex projects, communicating ideas, and collaborating effectively in team settings.

Students will not be graded on whether the diagram represents the "most efficient" order of operations or is perfectly optimized. Instead, the focus will be on clarity. The diagram should stand alone and be understandable without requiring additional written explanation.

Final Project

For their final projects, students will be asked to combine the components and archive they made throughout the semester into a working system.

Students will not be graded on whether their program represents the "most efficient" order of operations or is perfectly optimized. The focus will be on demonstrating an understanding of the core components of the system by submitting a working system.

Presentation

The presentation should demonstrate students' critical problem solving and their ability to communicate clearly about a system they designed.

Students should:

- Describe the problem the system solves (for instance, what insights does it enable).
- Describe their archive (overview of contents, sources, and significance).
- Describe their system components.
- Present the data visualization generated by their system.
- Offer a reflection: What did you like? What was challenging? What advice would you give others?

Participation

Participation in this course includes submitting assignments on time, arriving on time for each class session, and actively engaging in class discussions and activities. These elements are essential to fostering a collaborative and productive learning environment and will have an impact on your overall participation grade.

Attendance and Late Work

Attendance Policy

- No attendance will be taken the first week of class (since new people often enroll late)
- You may miss up to 4 classes with no grade penalty. No explanations are needed.
- Each unexcused, non-emergency absence after your allotted 4 absences will reduce the final grade by 1 point.
- Additional classes may be missed without grade penalty only for university-excused absences (e.g., religious holy days, jury duty, military service, officially documented illness or family loss,etc.) with formal documentation provided by Student Outreach and Support (SOS).

Tardies

If you are 5-20 minutes late to class, or leave class 5-20 minutes early, you are tardy. Three tardies are equivalent to 1 absence.

Absences

If you are 20 or more minutes late to class, or leave class 20 or more minutes early, this counts as being absent from class for the day.

Note: Canvas shows an “Attendance (Roll Call)” assignment but will NOT automatically factor attendance into your final grade shown. This will be done manually at the end of the Semester

Late Work Policy

Late homework is not accepted except in case of university-excused absences (e.g., religious holy days, jury duty, military service, officially documented illness or family loss, etc.). See information below on contacting Student Outreach and Support (SOS) to obtain documentation in support of requests for extensions on academic deadlines. SOS will verify details and provide documentation without revealing protected details. The Student Absence Notification Request Form is intended for critical situations and/or medical or family emergencies. In general, starting homework early reduces the chance that you will need an extension if someone unexpected should happen close to a deadline for submitting work.

TEXAS
The University of Texas at Austin

Academic Integrity

All students at UT Austin must abide by the Student Honor Code. This includes that: “I pledge to be honest about what I create and to acknowledge what I use that belongs to others.” Honor code violations harm yourself as well as other students, and this is strictly enforced. As noted below, students that violate University rules on academic dishonesty risk severe penalties, such as automatically failing the course and being dismissed from UT. Please do NOT take the risk.

Educate Yourself About Plagiarism. If in doubt, ask the instructor. Plagiarism and similar conduct constitute a serious violation of UT's Honor Code and standards of conduct.

Students who violate University rules on academic misconduct are subject to the student conduct process. A student found responsible for academic misconduct may be assigned both a status sanction and a grade impact for the course. The grade impact could range from a zero on the assignment in question up to a failing grade in the course. A status sanction can range from a written warning, probation, deferred suspension and/or dismissal from the University. To learn more about academic integrity standards, tips for avoiding a potential academic misconduct violation, and the overall conduct process, please visit the Student Conduct and Academic Integrity website.

When possible, assignments will be automatically checked for plagiarism. We report suspected cases to the central administration for investigation and disciplinary hearings

Class Collaboration Policy

Unless otherwise specified, it is the expectation of the University that you complete all work independently. For this class, if you discuss assignments with classmates, YOU MAY NOT take any written notes away from such discussions, unless explicitly directed otherwise. Any written notes from such discussions violate this collaboration policy, and could also lead to either accidental or intentional collusion in submitted work, in further violation of this collaboration policy.

Automatic Code Generation Tools are Prohibited (except for class time on April 10th)

Between you, me, and ChatGPT, I use AI nearly everyday. Nonetheless, automatic code generation tools are prohibited in this class, except for April 10th, during class time. The reason automatic code generation tools are prohibited is because the purpose of this class is to provide an introduction to **computational thinking** and to equip you with practical know-how in order to excel in a diverse and ever-changing technological landscape. You will have plenty of opportunities in the future to implement automatic code generation tools and AI in meaningful ways. **Importantly, to engage with AI meaningfully and critically, it is essential to have a strong foundational understanding of the specific fields or domains where the AI will be applied**.

Understanding and engaging the content in this course is key to acquiring foundational knowledge about the domain of Python programming and a deeper understanding of how **computational thinking** can be used to solve diverse problems, **even problems beyond programming**!

Sharing of Course Materials is Prohibited

Sharing of course materials is prohibited. No materials used in this class, including (but not limited to) lecture hand-outs, videos, assessments (quizzes, exams, papers, projects, homework), in-class materials, review sheets, or additional problem sets may be shared online or with anyone outside of the class without my explicit written permission. Unauthorized sharing of materials may facilitate cheating. The University is aware of the sites used for sharing materials, and any materials found online that are associated with you, or any suspected unauthorized sharing of materials will be reported to Student Conduct and Academic Integrity, as with other honor code violations above.

Safety and Wellness

Office of Emergency Management

- Sign up for Campus Emergency Text Alerts
- Emergency Preparedness Pocket Guide
- Emergency Terms (PDF)
- Remember 5 - Survival Guide (PDF)
- Active Shooter Response Guide

Coping with Stress and Personal Hardship

Life brings unexpected surprises to all of us. If you are facing any personal difficulties in coping with life challenges, please take advantage of the various services offered if they might help. They exist to be used.I invite students who are struggling for any reason and who believe that it might impact their performance in the course to reach out to me if they feel comfortable. This will allow me to provide any resources or accommodations that I can. If immediate mental health assistance is needed, call the Counseling and Mental Health Center (CMHC) at 512-471-3515 or contact Bryce Moffett, LCSW (iSchool CARE counselor) at 512-232-2983. Outside CMHC business hours (8a.m.-5p.m., Monday-Friday), contact the CMHC 24/7 Crisis Line: 512-471-2255.

Behavior Concerns Advice Line (BCAL)

If you have concerns about the safety or behavior of fellow students, TAs, or Professors, call the Behavior Concerns Advice Line (BCAL): 512-232-5050. Your call can be anonymous. If something doesn't feel right–it probably isn't. Trust your instincts and share your concerns.

Student Outreach and Support (SOS)

SOS provides assistance, intervention, and referrals to support students navigating challenging or unexpected issues that impact their well-being and academic success. If you need to miss class due to a family emergency, medical or mental health concern, or academic difficulty due to crisis or an emergency situation, SOS will verify your situation and notify your professors. You can best access services available through SOS by completing an online form or emailing them.

Basic Needs Security

Any student who faces challenges securing their food or housing and believes this may affect their performance in the course is urged to contact the Dean of Students for support. UT maintains the UT Outpost (https://deanofstudents.utexas.edu/emergency/utoutpost.php) which is a free on-campus food pantry and career closet. Furthermore, please notify the professor if you are comfortable in doing so. This will enable him to provide any resources that he may possess.

Further Information

University Resources. For a general list of university resources that may be helpful, see the University Resources Students Canvas page.

The Sanger Learning Center

Did you know that more than one-third of UT undergraduate students use the Sanger Learning Center each year to improve their academic performance? All students are welcome to take advantage of Sanger Center's classes and workshops, private learning specialist appointments, peer academic coaching, and tutoring for more than 70 courses in 15 different subject areas.

Disability and Access

Disability and Access (D&A). The university is committed to creating an accessible and inclusive learning environment consistent with university policy and federal and state law. Please let me know if you experience any barriers to learning so I can work with you to ensure you have equal opportunity to participate fully in this course. If you are a student with a disability, or think you may have a disability, and need accommodations please request them. If you are already registered with D&A, please deliver your Accommodation Letter to me as early as possible in the semester so we can discuss your approved accommodations and needs in this course.

Personal Pronouns

Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with differences of race, culture, religion, politics, sexual orientation, gender identity & expression, and nationalities. Class rosters are provided to the instructor with the

student's legal name, unless they have added a "chosen name" with the registrar's office, which you can do so here: https://utdirect.utexas.edu/apps/ais/chosen_name/. I will gladly honor your request to address you by a name that is different from what appears on the official roster, and by the pronouns you use (she/he/they/ze, etc). Please advise me of any changes early in the semester so that I may make appropriate updates to my records. For instructions on how to add your pronouns to Canvas, visit https://utexas.instructure.com/courses/633028/pages/profile-pronouns. More resources available on the Gender and Sexuality Center's website, www.utgsc.org.

UT Austin - Land Acknowledgement

(I) We would like to acknowledge that we are meeting on the Indigenous lands of Turtle Island, the ancestral name for what now is called North America. Moreover, (I) We would like to acknowledge the Alabama-Coushatta, Caddo, Carrizo/Comecrudo, Coahuiltecan, Comanche, Kickapoo, Lipan Apache, Ysleta Del Sur Pueblo and Tonkawa, and all the American Indian and Indigenous Peoples and communities who have been or have become a part of these lands and territories in Texas.

Course Schedule

The schedule outlined in this syllabus serves as an initial plan for the course. Adjustments may be made based on the pace of class discussions, the complexity of questions raised, or the specific needs of ongoing projects. Any changes will be communicated in advance to ensure students can plan accordingly.

Week	Date	Day	Topic	Due
1	Jan 14	T	**Class:** Introductions and syllabus **Reading:** N/A **Assignment:** Create a UTMail account	
1	Jan 16	TR	**Class:** Introduction to Python and Google Colab Upload data to Google Drive **Reading:** PY4E, Lesson 2, "Why Program?"	

			PY4E, Lesson 3, “Variables, expressions and statements” **Assignment:** Getting Started with Google Colab	
2	Jan 21	T	**Class:** Why program? Variables, expressions, and statements **Reading:** PY4E, Lesson 4, “Conditional Execution”	PY4E, Lesson 2, “Why Program?” PY4E, Lesson 3, “Variables, expressions and statements” Getting Started with Google Colab
2	Jan 23	TR	**Class:** Conditional Execution Decision Structures.ipynb **Reading:** PY4E, Lesson 5, “Functions” **Assignment:** Input -> Output	PY4E, Lesson 4, “Conditional Execution”
3	Jan 28	T	**Class:** Functions **Reading:** PY4E, Lesson 6, “Loops and Iterations”	PY4E, Lesson 5, “Functions” Input -> Output
3	Jan 30	TR	**Class:** Loops and Iterations. Diagram drawing tools (e.g. draw.io, Miro, Lucidchart). Setting up draw.io **Reading:** PY4E, Lesson 7, “Strings” **Assignment:** Draw a 'while' Loop	PY4E, Lesson 6, “Loops and Iterations”
4	Feb 4	T	**Class:** Strings **Reading:**	PY4E, Lesson 7, “Strings” Draw a 'while' Loop

TEXAS
The University of Texas at Austin

			PY4E, Lesson 8, “Files” “A Brief Introduction to the Python os Module” “A Brief Introduction to Google Colab Drive Integration”	
4	Feb 6	TR	**Class:** Files **Reading:** PY4E, Lesson 9, “Lists” **Assignment:** File Reader	PY4E, Lesson 8, “Files” “A Brief Introduction to the Python os Module” “A Brief Introduction to Google Colab Drive Integration”
5	Feb 11	T	**Class:** Lists **Reading:** PY4E, Lesson 10, “Dictionaries”	PY4E, Lesson 9, “Lists” File Reader
5	Feb 13	TR	**Class:** Dictionaries **Reading:** PY4E, Lesson 11, “Tuples” **Assignment:** Word Counter	PY4E, Lesson 10, “Dictionaries”
6	Feb 18	T	**Class:** Tuples Reusing code (modifying word_counter.ipynb so it imports file_reader()). **Reading:** Introduction to Pandas	Introduction to Pandas Word Counter
6	Feb 20	TR	**Class:** Dataframes **Reading:** PY4E, Lesson 12, Regular Expressions **Assignment:** Ngram Counter	

TEXAS
The University of Texas at Austin

7	Feb 25	T	**Class:** Regular Expressions **Reading:** "This is Not a String"	Ngram Counter PY4E, Lesson 12, Regular Expressions
7	Feb 27	TR	**Class:** Reading Errors Defining "Messy" An analysis of data cleaning packages (cautionary tales) **Reading:** PY4E, Lesson 15, "Object Oriented Programming" **Assignment:** Data Cleaner	"This is Not a String"
8	March 4	T	**Class:** Object-Oriented Programming (OOP) and Modularity **Reading:** N/A	Data Cleaner PY4E, Lesson 15, "Object Oriented Programming"
8	March 6	TR	**Class:** OOP and Refactoring Refactoring the Data Cleaner **Reading:** PY4E, Lesson 13, Network Programming **Assignment:** Refactoring for OOP	
9	March 11	T	**Class:** Network Programming **Reading:** N/A	PY4E, Lesson 13, Network Programming Refactoring for OOP
9	March 13	TR	**Class:** Finding and creating digital archives **Reading:**	

			PY4E, Lesson 14, Using Web Services **Assignment:** Creating an Archive	
Spring Break: March 17 - 22				
10	March 25	T	**Class:** Using Web Services **Reading:** PY4E, Lesson 16, Databases	PY4E, Lesson 14, Using Web Services Creating an Archive
10	March 27	TR	**Class:** Set Up SQLite with Google Colab and Drive **Reading:** N/A **Assignment:** Database Setup	PY4E, Lesson 16, Databases
11	April 1	T	**Class:** Databases **Reading:** N/A	Database Setup
11	April 3	TR	**Class:** Storing and Accessing SQLite Data **Reading:** PY4E, Lesson 17, Data Visualization **Assignment:** Store and Retrieve Data From Your Archive	
12	April 8	T	**Class:** Data Visualization System Diagrams vs. Flow Diagrams **Reading:** N/A	PY4E, Lesson 17, Data Visualization Store and Retrieve Data From Your Archive
12	April 10	TR	**Class:** Experimental Human-in-the-Loop AI Development Day: Creating a visualization	

			component human_in_the_loop_dev.ipynb Referencing online documentation for plotly **Reading:** N/A **Assignment:** Critical Human-in-the-Loop Reflections	
13	April 15	T	**Class:** System Integration and Troubleshooting Refactoring our code for OOP **Reading:** N/A	Critical Human-in-the-Loop Reflections
13	April 17	TR	**Class:** System Integration and Troubleshooting **Reading:** N/A **Assignment:** Finish presentations and final projects!	
14	April 22	T	**Class:** Presentations **Reading:** N/A	Presentations
14	April 24	TR	**Class:** Presentations **Reading:** N/A	Presentations Final Project: System Integration
Last day of classes, university-wide: April 28				