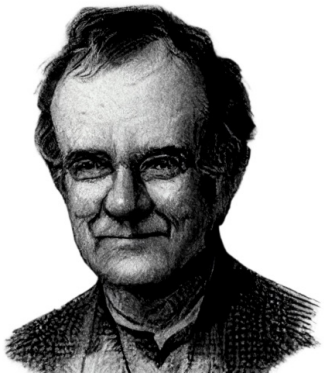
GEORGIA DOING

COURSE: CSC:XXX COURSE TITLE DOCUMENTATION

SELF

Contents

Home



The course website for CSCXXX, part of the Union College CS curriculum. Here you can find our wekkly schedule, assignmnets and resources.

Announcements

Upcoming due dates

Next assignments is due …

Office Hours

Course Description

Schedule

Weekly Schedule

Schedule

	Week	Topic	Assignment
1		topic…	assignment…
2		topic…	assignment…
3		topic…	assignment…
4		topic…	assignment…
5		topic…	assignment…
6		topic…	assignment…
7		topic…	assignment…
8		topic…	assignment…
9		topic…	assignment…
10		topic…	assignment…

Resources

Syllabus

Syllabus

CSC106 Syllabus

1 CSC 106: Can Computers Think?

- Union College, Fall 2025
- Georgia Doing, PhD: email: doingg@union.edu, office: Steinmetz 108B

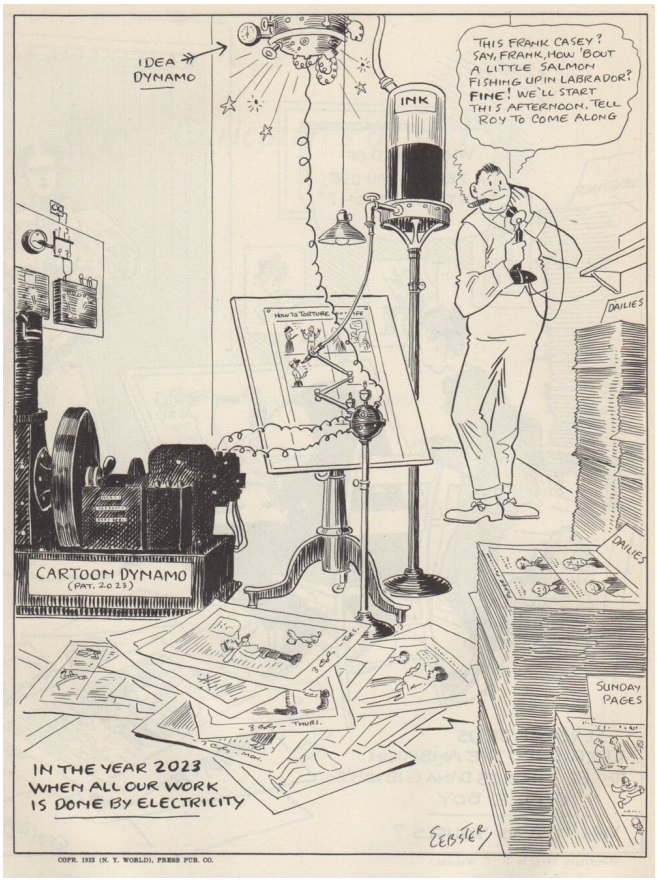


Figure 1: Cartoon by H.T. Webster published in 1922.

2 COURSE BASICS

- Lecture: Monday, Wednesday, Friday 11:45 am - 12:50 pm
- Lab: Thursday 10:55 am - 12:40 pm
- Location: Olin 107
- Student/Office Hours:
  - Wednesday 2:00 pm - 3:30 pm
  - Thursday 2:00 pm - 3:30 pm
  - drop-in or schedule a 15 minute slot: https://calendar.app.google/8bus6pfDvyphR9ar5
  - by appointment for another time or over zoom

3 OVERVIEW

Can computers think? Is it possible to build intelligent machines? What does it mean for a machine to think or to be intelligent? Why, or why not, would we want intelligent machines? To what extent do we already have them? How do they work? How do they impact our lives? What are their limitations? This course invites you to explore these questions and more.

This course is an introduction to the field of computer science with an artificial intelligence theme that does not assume you have prior experience with computer programming or computer science. It introduces basic algorithms, data structures and programming techniques. You will learn how computer scientists think about and solve problems by doing your own computer programming. This will give you an understanding of how the current technology for building intelligent machines works. In addition, you will read non-technical papers about artificial intelligence and the ethical questions raised by its use. You will reflect on the ways current computational systems often reinforce biases and can harm groups of people inequitably. And you will evaluate how the choices you are making when creating your own programs will affect future users.

If you're thinking about a CS or CPE major or minor, this course will give you a solid foundation to continue to *Programming on Purpose* (CSC 120), to *Data Structures* (CSC 151) and beyond. If you're a neuroscience major, it will help you see how scientists are using biological and neurological principles to model "behavior' ' in computers and it will teach you skills useful for processing experimental data. If you are thinking about another major or minor, this course will give you foundations to apply computing in whatever area you are interested in. And if you're here just because you're curious, that's great! This course requires the pre-approval of the department. For more information, please visit: https://union.edu/advising-registration/pre-approval.

4 LEARNING OBJECTIVES

By the end of the course, you should be able to *answer* the following questions:

- What are the most important programming fundamentals and why are they the building blocks of programs?
- What are the debugging fundamentals and how do you use them?
- What are some of the mechanisms used to build "intelligent" machines?

By the end of the course, you should be able to *do* the following:

- Use variables, functions, conditionals, loops and recursion
- Read Python programs
- Write short Python programs that solve a given problem
- Fix short Python programs using debugging fundamentals

Computing/programming topics we will cover include functions, variables, assignment, conditionals, loops, recursion, lists/arrays, file reading/writing, binary numbers and ASCII, strings and string methods and dictionaries.

A 1-page schedule-at-a-glance is included at the end of the syllabus, but please see Nexus for a link to a more detailed schedule.

5 CLASS POLICIES AND GUIDELINES

The Computer Science Department as a whole welcomes all people, regardless of age, background, beliefs, ethnicity, gender identity, gender expression, national origin, religious affiliation, sexual orientation and any other differences, be they visible or non-visible. It's also important to recognize that institutional racism has prevented members of marginalized groups - especially black, indigenous and other people of color (BIPOC) - from fully participating in the field of Computer Science.

You. Yes you, the one reading this syllabus. You belong here.

As an instructor, I will do my utmost to uphold these principles and to treat everyone with respect. As students in this class, I expect you to do the same since one person is not a community unto themselves. We're all in this together, so let's treat each other well. Lastly, I welcome feedback on issues we might discuss or ways that our class can be a more just one for all people.

6 COURSE MATERIALS

Online Textbook: Python for Everybody - Interactive (free).

We are going to use an online, interactive, free and open source textbook. This textbook has a lot of embedded interactive exercises. Reading this book always also means doing the activities. To get access to the book, please register on Runestone using the following informationm which can also be found on Nexus:

- https://runestone.academy/runestone/default/user/register
- your Union College email
- unioncollege_py4e-int_fall25 as the course name

We'll use the following websites and software throughout the course:

- Nexus: the course website
- Gradescope: used to submit programming assignments
- Python 3.13.X: the software package needed to write and run Python programs
- Reeborg's World: a useful website for learning some programming basics
- Python Tutor: a useful website for tracing python code

7 COURSE RESOURCES

The Nexus page has a list and links to the readings and software resources that we are going to use. In class, we are going to use the Linux machines in Olin 107. Outside of class you have a choice or hardware. You can install the software on your own computer (Python is freely available for Windows, Mac, and Linux, Reeborg's World and Python Tutor are browser based programs) or you can work in one of the Computer Science labs. We have three spaces that you can use 24/7 using your ID card, except when classes are being held in them:

- Olin 107 - where we have class
- PASTA Lab (ISEC 051+)
- Computer Science Resource room (Steinmetz Hall, 209A)

8 LIBRARY RESOURCES

The library is available to help you with your research needs! Librarians can help you develop research questions, search for and select the best sources for your projects, identify research strategies, evaluate sources, and assist you with creating citations. There are multiple ways for you to contact a librarian. For more information, please see our Ask A Librarian page: https://www.union.edu/schaffer-library/ask-a-librarian.

9 ACCOMODATIONS

It is the policy of Union College to make reasonable accommodations for qualified individuals with disabilities. If you are a person with a disability and wish to request accommodations to complete your course requirements, please make an appointment with me or stop by during my office hours as soon as possible, all discussions will remain confidential. You must provide reasonable notice and be in touch with Accomodative Services on the 2nd floor of Schaffer Library, if you have not already, if you wish to take advantage of extra time on exams.

10 COMPLEX QUESTIONS: GLOBAL CHALLENGES & SOCIAL JUSTICE

Justice, Equity, Identity, Difference (JEID). Algorithms and computational systems increasingly shape the world we live in. Advances in AI and machine learning are in the news on a daily basis. But there is also a growing awareness of the limitations and dangers of these systems. In particular, they often promote biases and inequalities. For example, facial recognition software has been shown to work much less accurately for people of color than for white people, an AI hiring algorithm was found to reject female applicants for technical positions at a higher rate than male applicants, and search results can be full of racial stereotypes. In this course, you will learn about real-world cases of AI systems that discriminate against groups of people. You will see that related ethical questions are raised even in the context of the relatively simple programs you create in an introductory computer science class. Additionally, you will work on developing a habit of always evaluating how your own choices as a programmer might impact other people's experiences.

Engineering, Technology and Society (ETS). In this course, you learn to break down a larger problem into a series of smaller computational steps (creating an algorithm) and to implement algorithms in Python. Through these activities you will gain an understanding of how computers work and an idea of the software development process. In particular, you will learn about iterative development and the importance of thorough testing. You will practice communicating about algorithms and their implementation in groups.

You will also explore, through assignments and discussion, how computational systems can have unintended and negative consequences for groups of people and what the responsibility of the programmer is to mitigate these dangers.

Data & Quantitative Reasoning (DQR). Much of current AI systems are data driven and learn based on collections of data. In this course, you will learn about and implement some approaches that AI systems use to draw inferences based on data. Many of the ethical issues around current AI systems are linked to the data these systems use. Throughout the course, you will, therefore, also reflect on: What information is collected? How is the information represented? Who is represented in the data? How is it used?

11 GRADING

The following components will contribute to your final grade for this class with roughly the indicated weight.

- 2 practical exams: 20% (10% each)
- Written midterm: 15%
- Written final exam: 15%
- Programming projects: 40%
- Engagement: 10%

Note: You must get a C- or better in order to take any other class that has a CSC-10x prerequisite. Several larger programming projects will be assigned to reinforce the concepts discussed in class and put into practice in homework assignments. All projects can be completed using programming techniques that have been covered in class.

Do not use techniques that have not been covered in class unless specifically told otherwise. Part of the challenge for each programming project is figuring out how to complete the assignment using only the tools provided in the course up to that point.

Letter Grade Scale:

- A: 93-100; A-: 90-92
- B+: 87-89; B: 83-86; B-: 80-82
- C+: 77-79; C: 73-76; C-: 70-72
- D: 60-69; F: 0-59

12 ATTENDANCE

You are expected to be present for every class. I realize that sometimes other things come up (interviews, athletic events, illness, etc.). In those cases, just let me know (in advance, if at all possible) that you are going to be absent. However, you should not miss more than four class sessions, which would amount to about 10% of the course. No matter how much class you have missed, please do not come to class if you have an illness that is likley contageous, please email me to work out a reasonable solution.

If you do miss class, it is your responsibility to make up any material that you missed. Get notes from a classmate, make sure to complete all homework assignments due or assigned during the class that you missed, and come see me if you have any questions on the material. Unless you have made an arrangement with me ahead of time or you had an emergency, I will expect you to hand in all assignments by the due date, regardless of whether you were in class. You will not be able to make up missed exams or in-class questions and assignments.

13 LATE POLICIES

Homework assignments will usually be discussed in class on the day that they are due, but will not be individually graded. Programming projects are due by 11:59:59pm on the due date. You have three "extension tokens" for this class. Each token will extend a single project deadline by 24 hours. You can use each token on a different project, or use two or even all three on a single project. Once all three tokens have been used, no late submissions will be accepted (barring exceptional circumstances).

14 ENGAGEMENT

We will use lab notebooks throughout this course. Lab notebooks can be used as your primary notebook for lecture notes or in addition to your preferred notebook style. These hand-written, real-time records of your thoughts and thought processes are critical parts of problem solving and also serve as evidence of individual student engagement to augment participation in class discussion, group activities and weekly homeworks. To maximize their utility, lab notebooks will be periodically reviewed and discussed. If you have concerns about this policy, please let me know, I want to ensure that this policy supports our work while meeting your needs as a student. Following these guidelines will constitute postiive engagement:

- Respect. This course is a space for rigorous and respectful debate. When confronting ideas and people different from you, lead with curiosity rather than judgement. This commitment extends to all our face to face and digital interactions. We will critique ideas, not people. To protect our shared learning environment, you may not record, photograph, or share any part of our class sessions outside of our class.
- Show up to class on time and prepared. Show up ready to learn by completing the readings and exercises. When you are late or unprepared, you are disrespecting the learning experience for the group.
- Embrace intellectual humility. Recognize that there are limitations to your knowledge and that some of your beliefs could actually be wrong. Be curious about your thinking and open to learning from others. This is hard, but so vital to your learning in this course—we'll work on developing this throughout the term!
- Get help when you need it. If you are stuck or confused or lost, be proactive and get help. The best learners and thinkers can figure out when they are stuck and make a plan to get unstuck. Come to Office Hours (Wed/Thurs 2:00-3:30pm Steinmetz 108B), the CS Helpdesk (Sun-Thurs 7:00-9:00pm Olin 107) or ask another student. Often the best person to explain something is the person who just figured it out. Try reaching out to another student in the course who seems like they get it—they will probably be flattered you asked!

15 ACADEMIC INTEGRITY AND "ARTIFICAL INTELLIGENCE" USE

Union College recognizes the need to create an environment of mutual trust as part of its educational mission. Responsible participation in an academic community requires respect for and acknowledgement of the thoughts and work of others, whether expressed in the present or in some distant time and place. Matriculation at the College is taken to signify implicit agreement with the Academic Honor Code, available at: http://muse.union.edu/honorcode

It is each student's responsibility to ensure that submitted work is their own and does not involve any form of academic misconduct. Students are expected to ask their course instructors for clarification regarding, but not limited to, collaboration, citations, and plagiarism. Ignorance is not an excuse for breaching academic integrity. Students are also required to affix the full Honor Code Affirmation, or the following shortened version, on each item of coursework submitted for grading:

'I affirm that I have carried out my academic endeavors with full academic honesty.'

[Signed, Jane /John /Jaden Doe]

What it means for this course: **you must complete the programming projects on your own, but you are welcome and encouraged to collaborate on other regular assignments.**

Please notice that different rules apply to the exercises that are part of the textbook, regular assignments and programming projects.

Textbook Readings and Regular Exercises:

You are welcome to discuss the textbook readings, the exercises embedded in the textbook, and other regular assignments with other people in the class to help you fully understand the material. However, don't just copy somebody else's solution. Even if you ask other people for help, the goal is for you to understand the underlying concepts and logic. It's more important to understand how to arrive at a correct solution than it is to know the solution itself.

Programming Projects:

Any work you turn in has to be your own. The best way to learn programming is by doing programming. I expect that you already know that if you do not do things for yourself, you will not learn them. Sometimes that will mean that you will have to struggle. That is ok: struggling to figure out how to solve a problem is where a lot of learning happens. Give yourself the opportunity to do this.

Please, do not ever copy any part of a solution from a friend, from any material generated by an "artificial intellence" (AI) chatbot model (ChatGPT, Gemini, Copilot, Claude, etc.) or from anywhere else. Do not ever *look at* somebody else's solution or any solution generated by an entity other than yourself before you have completed your own. Do not ever *show*

somebody else your solution before you've both submitted it; though it may be beneficial to discuss your solution with another student in the class after you've both handed in the assignment. Please do not *ever* show your solution to future students of this class or any chatbot, and do not *ever* post your solutions on the Internet.

Don't search for information on the Internet. Don't ask for information from any AI model. Especially, don't look at any material that provides solutions to exercises that are similar to the projects. An exception is, of course, any online material that is linked directly from the Nexus page. As always, you should practice good citation behavior and cite any sources that you consult in your work (for example, which pages from the Python documentation website you consulted when working on the project).

Discussing your work (without looking at code): Talking through a problem is extremely beneficial in understanding the problem and coming up with a solution path. So, I strongly encourage you to talk to other students in this class about the projects, visit the CS Helpdesk and come to my Office Hours (you can also request a meeting if you have schedule conflicts with my Office Hours).

Here are some ways in which you can talk about the projects without having to worry about violating the rules from above.

- It's ok (and strongly encouraged!) to draw **memory diagrams** together. Draw memory diagrams and discuss possible algorithms for solving the problem.
- It's ok to **write down an algorithm in English** together. Then go off by yourself to implement it into code.
- It's ok to read and write code together that is not part of the assignment, but that helps demonstrate the concept of what you're being asked to do. Go through an example from class or from the book together, or **come up with your own examples**. There's no better way to understand something than trying to think up examples in order to teach it to someone else. Try it!

Do not work on the implementation together with friends, especially if you are each at your own computer and are always in lock step trying to figure out the same line or section of code at the same time. Have a higher-level discussion (using one of the strategies suggested above), then work on the implementation on your own.

Your goal for discussions about any assignment should be that you come away with a better understanding of the problem and of possible ways to approach it so that you can then try out these approaches on your own. You should never leave a discussion with just an answer, without an understanding of how to arrive at that answer. A good general guideline for any discussion, or interaction with an AI model, is that you should not leave the discussion with anything written or typed.

16 CONTACTING ME OUTSIDE OF CLASS

The best method for contacting me outside of class is to stop by my office during Office Hours or whenever my office door is open. You can also schedule a time with me if my regular office hours don't work. Please contact me via email (doingg@union.edu) and we can arrange a phone or zoom call. I respond to emails as soon as possible, but it may take up to one day to get a response during the term, and longer between terms.

17 ADDITIONAL RESOURCES

Mental Health and Campus Resources

Union College is committed to supporting and advancing the mental health and well-being of our students. During the course of their academic careers, students often experience personal challenges that contribute to barriers in learning, such as drug/alcohol problems, strained relationships, chronic worrying, persistent sadness or loss of interest in enjoyable activities, family conflict, grief and loss, domestic violence, difficulty concentrating, problems with organization, procrastination and/or lack of motivation. Students also sometimes come to college with a history of learning difficulties (e.g., any form of special education), experience difficulties succeeding in a particular subject (e.g., math, reading), or have experienced some form of trauma be it emotional or physical (e.g., head injury). These mental health concerns can lead to diminished academic performance and can interfere with daily life activities. If you or someone you know has a history of mental health concerns or if you are unsure and would like a consultation, a variety of confidential services are available. The Eppler-Wolff Counseling Center provides free counseling and therapy services (including psychiatry) to all enrolled students. Please call (518) 388-6161 or visit the Wicker Wellness Center in person any weekday between 8:30 and 5:00 to schedule an initial contact appointment. Visit the Counseling Center website for more information.

In a crisis situation, or after hours, contact Campus Safety at (518) 388-6911. The National Suicide Prevention hotline also offers a 24-hour hotline at (800) 273-8255.

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 (dos_office@union.edu) for support or drop into office hours Monday - Friday 8:30 am - 5:00 pm in the Reamer Campus Center room 306. Furthermore, please notify me if you are confortable doing so and I will provide any resources I can. The Union College Persistance Fund can provide financial support in times of unexpected harship to cover needs such as emergency medical expenses not covered by insurance and basic living expenses. Additional sources of supoprt are outlines on the Dean of Students webpage: https://www.union.edu/dean-students .

17.1 ADDENDA

Any community agreed upon additions:

18 TENTATIVE SCHEDULE

W	TOPIC	Due	Runestone
1	Programming, Calling Functions	Introductory Survey	2.1 - 2.13
2	Loops, Variables and Assignment	HW 1	3.1-3.5
3	Defining Functions	HW 2, Project 1	5.1-5.12
4	Conditional Statements	HW 3, Practical Exam 1	4.1-4.9
5	Conditional Loops	HW 4, Project 2	6.1-6.7
6	Strings, Lists	HW 5, Written Exam 1	7.1-7.12
7	Nested Lists	HW 6, Project 3	9.1-9.14
8	File I/O, Dictionaries	HW 7, Project 4	8.1-8.9
9	Dictionaries	HW 8, Practical Exam 2	10.1-10.6
10	Searching, Sorting, Recursion	HW 9, Written Exam 2	

19 PROGRESS TRACKER

Learning Goals	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10
What are the most important programming fundamentals and why are they the building blocks of programs?										
What are the debugging fundamentals and how do you use them?										
What are some of the mechanisms used to build “intelligent” machines?										
Use variables, functions, conditionals, loops and recursion.										
Write short Python programs that solve a given problem.										
Fix short Python programs using debugging fundamentals.										

Grading

Table 2: Grading Scheme

Course Element	Grade Point Contribution	Notes
Final Exam	X%	written
Midterm Exam	X%	written
Project	X%	written
Engagement	X%	written
total	100%	

Table 3: Letter Grade Scale

Letter Grade	Percent range
A	93 – 100
A-	90 – 92
B+	87 – 89
B	83 – 86
B-	80 – 82
C+	77 – 79
C	73 – 76
C-	70 – 72
D	60 – 69
F	0 – 59

Assignments

Project

Weekly Notes

Week 1 Test

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

Week 8

Week 9

Week 10

about.qmd

References

Acknowledgements

Many thanks to the Quarto team for creating this wonderful tool. I am also grateful for the support I got at Quarto's Discussion Board, specialy:

- Mickaël Canouil (`@mcanouil`);
- Charles Teague (`@dragonstyle`);
- Raniere Silva (`@rgaiacs`); and
- Christophe Dervieux (`@cderv`)