

CSC111-INTRO TO PROGRAMMING

Spring 2025

Instructor:	Qixin Deng	Email:	dengq@wabash.edu
Office:	Goodrich Hall 108	Office Hour:	M/W/F Afternoon
Time:	T/Th 9:45 AM - 11:00 AM	Place:	Hays Science, 003 Lecture

Office Hours:

I am free most of the time on M/W/F afternoon before 5:00 pm. If you want to visit my office at another time, I hope you can make an appointment with me via email. You can email me at any time, I usually reply to emails very quickly.

Course Description:

We will use Python to learn computer programming. This course will start by introducing Python's basic syntax, variables, built-in data structure, flow control, and function definition. After you can write programs using Python, object-oriented programming will be introduced. Object-oriented programming (OOP) is a style of programming characterized by the identification of classes of objects closely linked with the methods (functions) with which they are associated. It also includes ideas about the inheritance of attributes and methods.

Reference Note:

1. The best way to learn a programming language is by using official documentation and tutorials.
2. Official Tutorial: <https://docs.python.org/3.7/tutorial/index.html>
3. Official Documentation: <https://docs.python.org/3.7/reference/index.html>
4. Stackoverflow: <https://stackoverflow.com/>

Platform:

1. Pycharm: <https://www.jetbrains.com/pycharm/> (Free for students!)
2. Google Colab (Online): <https://colab.research.google.com/Links>
3. Jupyter Notebook (Online): <https://jupyter.org/try-jupyter/retro/notebooks>

Course Goal:

- Learn basic Python syntax and grammar, basic data structures
- Learn control flow and functions
- Understand Object-Oriented Programming (OOP) and programming using OOP

Assignments:

You will have weekly assignments. For each of the assignments, you need to not only write the code but also clearly explain how you solved this question, which includes all the ideas you came up with and the mistakes you made while solving the problem. For example, you forgot to write the colon at the end of the for loop, or analysis of logical errors leading to incorrect output results, etc. You need to submit both code and a short report. **NO LATE SUBMISSIONS WILL BE ACCEPTED!**

Grading:

- 50% Assignments, 50% Exams
- I will apply the letter grade criteria as below:

Letter	A	A-	B+	B	B-	C+	C	C-	D+	D	D-
Total	>=93	>=90	>=87	>=83	>=80	>=77	>=73	>=70	>=67	>=63	>=60

From the college side, there are no D+ or D-, which means if you are below 70, your grade will be D or F.

Class rules:

The student who shows discrimination, disrespect, bullying, and other bad behaviors during the lecture will immediately fail the course. And I will report to both the department and the college.

Important Dates:

Midterm	TBD
Final	TBD

About AI:

Right now, AI is pretty powerful. Reasonable use of AI can make it a very good assistant for you. However, the misuse of AI will dumb you down. Do not try to let AI do your assignments for you, and even less so, let AI replace your thinking. If you do so, you will be replaced by AI in the future. You should rule AI, not let AI dominate you.

You must complete the assignment independently. If there is a significant discrepancy between your exam results and the assignment, I will require you to independently redo the assignment or a very similar set of problems in my office within a reasonable time frame. If you cannot provide satisfactory answers, or if the style and methods of the code you write differ greatly from what you submitted in the assignment, I will report you to the college.

Academic Credit Policy

This course complies with the [Wabash College Academic Credit Policy](#) stated in the Academic Bulletin.

Because we have very limited class time, students are required to learn the programming platform, read extra notes, and answer extra questions related to the notes.

- Each week, students will be assigned notes to review along with 5 additional questions designed to assess their understanding of the notes. Completing these tasks should take approximately 1 hour and will account for **12 additional 4th hours sessions** for the semester.
- Students will also be required to set up the programming environment and familiarize themselves with the platform used in class. This will take around 3 hours and will account for **3 additional 4th hours sessions** for the semester.

In total, the course includes 15 hours of additional structured activities outside of regular class meetings (equivalent to 1 hour per week for 15 weeks.)

Guidelines for Ethical Computing and Working with Others

For the purposes of this course, adhering to the Gentleman's Rule means also adhering to the ACM Code of Ethics and Professional Conduct. The Code lays out the fundamental ethical principles that should guide all students and professionals in their study and work when computing and technology are involved.

One major part of ethical computing is respecting the ideas and works of others, and only using them in an ethically appropriate and responsible manner. Computing, like almost all human endeavors, is a collaborative enterprise. We learn from each other and from the work of those who came before us, just as our descendants will learn from us. To not respect the works of others damages one's own learning, and ultimately damages human society as a whole.

For the purpose of this course, I will add the following guidelines. In general, unless specifically authorized for a particular assignment, you should refrain from providing program code or solutions to another student, or receiving program code or solutions from another student. More concretely: providing code or solutions to another student or receiving code or solutions from another student, unless specifically authorized, constitutes academic dishonesty and a violation of the Gentleman's rule. This applies also to both past and future students in this course.

For the purpose of this course, I am happy for students to work in collaborative partnerships. You can discuss it together, but you have to finish your exercises by yourself! Collaborative means learn from each other, do not copy-paste! You should not turn in anything unless you fully and completely understand it and can fully and completely explain it.

In order to foster your ability to communicate about technical matters, any assignment may include an interview with me, during which I will ask you to explain your code or solution and the process by which you came to it. While I acknowledge that technical communication is a skill that must be practiced to be improved, exceptionally poor performance during an interview will raise my suspicions that you may be in violation of the Gentleman's Rule and the ACM Code.

The Syllabus as a Living Document

A syllabus is a living document and can be amended. I will communicate any syllabus changes; it is your responsibility to be aware of course policies at all times.

Quantitative Skills Center (QSC)

Is biology shredding your brain into fine particulate matter? Does chemistry have your anxiety level reacting negatively? Do you have bugs in your computer science code that you just can't squash? Are the demands of your economics homework far outpacing your supply? Is math leaving you feeling irrational or derivative? Has your last physics exam, not gravity, been pulling you down? If you have questions about biology, chemistry, computer science, economics, mathematics, or physics, visit the Quantitative Skills Center (QSC)! Our department-selected tutors will work with you to answer your questions and deepen your understanding of each subject. The QSC operates Sundays, Tuesdays, and Thursdays from 7- 11pm on the second floor of the Library. Please check out the QSC webpage <https://www.wabash.edu/ace/qsc> for specific tutor availability. If you have any questions, please email Vic Lindsay at lindsayv@wabash.edu.

Writing Center

Do you have questions about how to start a paper? Are you struggling to get all of your ideas to fit? Do you have a draft but want someone to review it? Are you unsure of how to incorporate comments from your professor? Head to the Writing Center! No matter what your writing questions or needs, the Wabash Writing Center. Consultants are eager and able to help you! Please visit the webpage, <https://www.wabash.edu/ace/writing> to schedule an appointment. If you have any questions, please email the director, Dr. Koppelman, at koppelmz@wabash.edu.

The Office of Student Enrichment

Succeeding at Wabash College takes a great deal of effort and planning. Life is complex, assignments are time consuming, and staying involved keeps you running. When you have questions about how to make everything fit into your schedule, how to study more efficiently, how to take better notes, or any other question about developing your college skills, visit the Office of Student Enrichment (OSE). Go to <https://www.wabash.edu/ace/office> and follow the "Make an Appointment" link to arrange a one-on-one, personalized meeting with Dr. Koppelman or one of the OSE Advisors. No matter your questions,

we will work with you to find a solution that helps you achieve your goals.

Disability and/or Special Accommodations

Students with disabilities (apparent or invisible) are invited to confidentially discuss their situation with the disability coordinator, Heather Thrush, Associate Dean for Student Engagement and Success. If a student wishes to receive an academic accommodation, it is required that his documentation of the disability be on file with Dean Thrush, who can, in confidence, provide information and guidance. Early notification helps us all work together in the most effective ways. To make an appointment with Dean Thrush, call (x6347), or email thrushh@wabash.edu. Appointments can be available via Zoom, if needed.

Course Content Arrangement

This is a tentative schedule, we will not strictly follow the plan. As you know, there are 14 weeks of classes in total, here I only list class content arrangement of 13 weeks. I left some redundancy in order to make room for exams or some unexpected events.

Course Content Arrangement	
Week #	Course Content
Week 01	Introduction; Basic Syntax; Binary and Hex; Operators; Operator Precedence;
Week 02	Copy and Reference; Basic Input and Output
Week 03	Number; String; List
Week 04	Tuple; Set; Dictionary; If conditional statements
Week 05	For Loop; While Loop; Comprehension Statement; Iterator; Generator
Week 06	Function; Arguments; Return;
Week 07	Anonymous Function; Higher Order Function; Scope
Week 08	File Operations; Errors and Exceptions; Introduction to OOP
Week 09	Objected Oriented Programming: Abstraction and Encapsulation
Week 10	Objected Oriented Programming: Abstraction and Encapsulation
Week 11	Objected Oriented Programming: Inheritance
Week 12	Objected Oriented Programming: Polymorphism
Week 13	Decorators