CS65: Introduction to Computer Science (Fall 2025)

Instructor: Md Alimoor Reza

Assistant Professor of Computer Science
Department of Mathematics and Computer Science, Drake University
Office: Shuang-Zi Building 11^{th} Floor, Qingdao University

Logistics: Most of the course content lectures, assignments, labs, etc. will be hosted at following website: https://analytics.drake.edu/ñeza/teaching/cs65_fall25. Coursework will include attending the class and completing individual and collaborative programming assignments, labs, class activities, quizzes, and a final project.

Class meetings: Please refer to my course website for the most up-to-date schedule and URLs. The time slots for the three sections are as follows:

- **Section#1**
  - Time: Tuesday 02:00 pm - 03:50 pm (week 1-9)
  - Time: Thursday 10:10 am - 12:00 pm (week 1-15)
  - Location: Boyi building#510
- **Section#2**
  - Time: Tuesday 04:00 pm - 05:50 pm (week 1-15)
  - Time: Friday 02:10 am - 12:00 pm (week 1-9)
  - Location: Boyi building#510
- **Section#3**
  - Time: Tuesday 02:00 pm - 03:50 pm (week 1-9)
  - Time: Friday 08:00 am - 09:50 pm (week 1-15)
  - Location: Boyi building#510

Office hours: The instructor will hold weekly office hours in person. Please see my course website for up-to-date times and URLs. If you cannot make it during office hours, contact the instructor to make a separate appointment.
Section#1:

- Time: Tuesday 08:00 am - 09:50 am
- Time: Thursday 02:00 pm - 03:50 pm
- Location: Shaung-Zi Building 11^{th} floor, Qingdao University

Course overview: This is an introductory course to the field of computer science. I will broadly overview logical thinking and problem-solving with a high-level programming language (Python). A crucial component in problem-solving is called "algorithm," which is step-by-step instructions to be executed by the machine. Besides algorithms, this course will also focus on other important concepts such as data structures, control structures, debugging, etc. As we will be using a high-level language (Python) to implement the learned concepts, here are some specific programming topics (tentatively and not necessarily in this order):

- Variables and expression
- Recursion
- Files
- Tuples
- Exceptions
- Functions
- Loops
- Lists
- Class
- Conditionals
- Strings
- Dictionaries
- Inheritance

Learning outcome: After successfully taking this course, you will be able to:

- Understand fundamentals of computer science.
- Think and devise a step-by-step process for problem-solving.

- Understand and implement algorithms with useful data structures using Python programming language.
- Refine programs by debugging and testing to ensure their desired outcome.

Prerequisites: This is an introductory course. Four years of high school mathematics or MATH 20. Logical thinking and basic familiarity with computers and their applications are helpful.

Textbooks and materials: There is no required textbook for this course. Optional readings will come from the followings:

- *A Byte of Python by Swaroop*
  - free copy available online at https://python.swaroopch.com/.
- *Think Python 2nd Edition by Allen B. Downey*
- *A Practical Introduction to Python Programming by Brian Heinold*
- *Python for Everybody by Charles Severance*

Grading and requirements:

- *Homework Assignments (20%).*
  - homework Python programming activities (3-4 in total).
- *Programming Labs (35%).*
  - A series of short exercises accompanying nearly each class topic. Relatively easier than homework assignments (10-12 in total).
- *Content Quizzes (30%).*
  - "paper-and-pencil exams"quizzes based on the lecture contents. (4-6 in total)
- *Final project (10%).*
  - Individual proposed project. You also need to prepare a powerpoint presentation by the end of semester.
- *Attendance (5%).*
  - Counted based on your signature.

Grading scale: The tentative grading scale for this course would be as follows (subject to change upon Instructor's discretion):

• A+ (97%-100%)	• A (91%-96%)	• A- (83%-90%)
• B+ (77%-82%)	• B (73%-76%)	• B- (69%-72%)
• C+ (66%-68%)	• C (63%-65%)	• C- (62%)
• D+/D (61%)	• D- (60%)	• F (55%)

Attendance policy: Attendance is mandatory. Students are required to attend all weekly meetings. If you are unable to attend the class in person, please email the instructor.

Deadline policy: I believe it is important for students to keep up with all course material since the class moves quickly, and it is easy to get behind. Our policy on deadlines tries to balance these two competing considerations. If you can't meet the deadline for some valid reason, please get in touch with the Instructor.

Policy on ChatGPT usage: You are tentatively permitted to use ChatGPT and other AI-based tools to assist with your learning, including in the development of code you submit for labs, assignments, and in solutions to exam questions, under the following conditions:

- You must not present AI-generated work as if it were your own.
- You must fill out and submit the AI-Assisted Learning Reflection before the assignment's due date.
- You can not use ChatGPT (or any other AI assistance) on the content quizzes (but you can use anything accessible via Fanya such as readings or your own code)

Technology requirements: You will likely also want a Python and/or C++ development environment on your local computer. I recommend **Thonny Python IDE for beginners**`https://thonny.org`. There are many other free and commercial options; you can choose whichever you like. For example, see `https://wiki.python.org/moin/IntegratedDevelopmentEnvironments` for a list; popular choices include PyCharm (which is free for academic use), IDLE (a simple IDE that comes built-in to most Python distributions), and XCode (which is free but available only for Macs).

Collaboration and Academic Integrity Policy: *We take academic integrity very seriously.* You are required to abide by the Drake University policy on academic integrity, as described in the Statement on Academic Dishonesty: Cheating and Plagiarism (`https://www.drake.edu/studentlife/handbook-resources/handbook/academic/`). It is your responsibility to understand these policies. Students agree that by taking this course, papers and source code submitted to us may be subject to textual similarity review, for example, by Turnitin.com. These submissions may be included as source documents in reference databases solely for the purpose of detecting plagiarism of such papers or codes. Here is a list of items to consider about academic integrity policy for CS 65:

- You may work together on labs
- You may not work together on projects and quizzes
- When working together, say who you worked with in the code comments
- When taking code from somewhere on the Internet, cite it in your comments and provide a link
- May not look at code from previous students in the course
- May not search the Internet for solutions
- First violation: **0** on that thing
- Second violation: **F** for the course

Accommodation for students with disabilities: Drake University is committed to providing equitable access to learning opportunities for all students. The Disability Services office (107 Old Main) collaborates with students who have disabilities to provide and/or arrange reasonable accommodations. If you have, or think you may have, a disability (e.g., mental health, attentional, learning, autism spectrum disorders, chronic health, traumatic brain injury and concussions, vision, hearing, mobility, or speech impairments), please contact:

- Michelle Laughlin, Student Disability Services Coordinator (x1835)
- michelle.laughlin@drake.edu

to arrange a confidential discussion regarding equitable access and reasonable accommodations.

Holiday Observance: If you miss class because of a holiday or observance, you can fill out the form to automatically notify me (faculty). You can find the form on the `https://www.drake.edu/diversity/initiatives` for Initiatives and Programs, or click here to directly access the `https://drake.qualtrics.com/jfe/form/SV_d5qfVUKtuTQdg7b`.