

CS 124: Honors Introduction to Computing

Fall Semester 2023

<https://www.cs.clarku.edu/courses/cs124/>

Class Meetings:

MWF 12:00 – 1:15pm MACD117

John Magee, Instructor

jmagee@clarku.edu

Office hours to be scheduled and posted online

Please use Piazza for all course-related communication instead of email!

Course Description

Develops computational problem-solving skills through programming, and exposes students to a variety of other topics from computer science and its applications. The focus of the course is to learn fundamental computational concepts (information, algorithms, abstraction and programming) that are central to computer science, and that also happen to be instrumental for the computational investigation of science. Design, analysis, and testing of problem-solving techniques are applied to a variety of domains across the sciences and liberal arts. This is the first course for computer science majors and anyone seeking a rigorous introduction. No prior knowledge of programming is required, but good analytical skills are helpful.

Aim 1: To give students the tools to take a computational problem through the process of design, implementation, documentation, and testing.

Objectives:

- Break a broad problem down into specific subproblems
- Write an algorithm to solve a specific problem, and then translate that algorithm into a program in a specific programming language (Python)
- Write clear, concise documentation
- Develop test cases that reveal programming bugs

Aim 2: To give students an understanding of the breadth of Computer Science as a discipline and how it exists in the world.

Objectives:

- Identify applications of computer science in society
- Describe the big questions in computer science
- Describe the relationship between major sub-disciplines within computer science, including functional and imperative programming, computer architecture, and theoretical computer science

Teaching/Learning Method

Topics will employ a *flipped-classroom* model.

You will be responsible for content-acquisition before coming to class, and we will use in-class time for learning through active engagement and discussion.

For each topic, your **preparation** before coming to class will include:

- Read the relevant textbook sections to be introduced to vocabulary, concepts, and examples.
- Watch one or several short videos that present examples relevant to the topic.
- Re-create and submit the examples on your own, solving any syntax or logical issues you encounter. Bring your questions about the examples for discussion at the start of class.
- Take a short online comprehension check on Moodle to prove to yourself that you have acquired the basic concepts, vocabulary, and syntax.
- Read the details of the problem set posted online.

Your preparation before class is crucial to your success on the assignments.

In-class discussion will employ a *peer-instruction* approach.

The peer instruction approach will involve using the Poll Everywhere response system (“clicker questions”). You will be asked to answer questions during class and respond with the Poll Everywhere app. You will then “turn and talk” to your neighbors (or discuss in a Zoom breakout room) to work through a problem or gain further understanding of the topic before voting again with the app. The “clicker questions” will give you immediate feedback on your understanding of the material. Explaining concepts to your peers helps solidify your own understanding: the best way to learn something is to teach it!

This course will follow a **rigorous schedule of assignments**. Each assignment corresponds to a section of the course content and textbook. Failure to keep up with the assignments will likely result in unsatisfactory performance in the course. Each session builds on the prior session and is a required building block for the following session. It is very difficult for you to be successful in the course if you miss class sessions or assignments. If you start to fall behind, please seek immediate assistance to get caught up.

What You Need to Know About Computer Programming

We believe anyone can succeed at learning to program and to think like a computer scientist. This is a first course in computer science, and there are no formal pre-requisites. The only expectation of students' computer skills before taking this class is to be comfortable with using email, web browsing, and copying and pasting text. In addition, familiarity with high school-level algebra is assumed.

In addition, you will need time, and this is more important than you can imagine. Many people believe that computer programming is extremely difficult, and that the code is written in some arcane syntax understandable only by experts. Although some parts of the process are indeed complex, most of the source code required for homework assignments can be easily understood.

So, what makes programming so hard? It's not the difficulty: It's the time required to achieve any decent results. The homework assignments will take time, so make sure you have plenty of it.

Adapted from text in "Core Techniques and Algorithms in Game Programming," Daniel Sanchez-Crespo Dalmau, and Aaron Stevens (BU)

LEEP Learning Outcomes

The goal of this course includes introductory level mastery of LEEP Learning Outcomes as follows.

- 1. Knowledge of the Natural World and Human Cultures and Societies.**
Demonstrated through understanding of technical fundamental knowledge areas of computer science, applications of computer science to nearly every aspect of modern life, interaction of computing with many different fields, and the ability to communicate and learn from experts in other fields.
- 2. Intellectual and Practical skills**
Demonstrated through the ability to design and improve systems using quantitative and qualitative assessments of their functionality, usability, and performance; practice and improvement of communication skills and the ability to communicate effectively to a range of audiences about technical problems and their solutions; the ability to think at multiple levels of detail and abstraction, and apply that to construct and analyze systems; and a recognition of the context of the function of computer systems, including interactions with people and the physical world.
- 3. Personal and Social Responsibility**
Demonstrated through the recognition of the social, legal, ethical, and cultural issues inherent in the discipline of computing; an understanding individual and collective responsibility and the possible consequences of failure on real people or the world; a commitment to life-long learning, recognizing that the field of computing advances at a rapid pace; and the ability to manage one's own learning and development, including managing time, priorities, and progress.
- 4. Ability to Integrate Knowledge and Skills**
Demonstrated through the recognition of recurring themes and general principles that have broad applications in computing beyond the domains in which they are introduced; an understanding of the fundamental interplay between theory and practice in computing; the ability to solve problems by constructing complex systems, which are analyzed on multiple levels of detail and abstraction; and an appreciation for interdisciplinary aspects of computer science and the application of knowledge toward solving cross-disciplinary problems.
- 5. Capacities of Effective Practice**
Demonstrated through the application of students' knowledge toward solving real problems across a variety of domains; the pursuit of an integrative large-scale project that applies knowledge across the curriculum; the effective presentation of projects, ideas, technical problems, and solutions to a variety of audiences; the ability to work effectively in a team and enhance interpersonal skills through collaborative efforts; and the ability to recognize how to apply foundational knowledge to learn and adapt skills throughout a career, as specific technologies change over time.

Books/Resources

The main online textbook for this course is:

How to Think Like a Computer Scientist: Interactive Edition

Available free at: <https://runestone.academy/runestone/books/published/thinkcspy/index.html>

Optional Supplemental material:

CS for All

By Christine Alvarado, Zachary Dodds, Geoff Kuenning, Ran Libeskind-Hadas

ISBN 1590282906

Required Clicker Device:

iClicker 2 – Response Device needed for in-class questions.

Other online readings and tutorials will be posted to the schedule/Canvas page.

Course Credits: Course materials have been adopted from Alvarado, Dodds, Keunung, Libeskind-Hadas and colleagues at Harvey Mudd College and from David Sullivan and Aaron Stevens at Boston University.

Software

For the applied parts of the course, we will be using the following software.

- Python 3 – We recommend installing the Anaconda Python distribution <https://www.anaconda.com/distribution/>
- Visual Studio Code: <https://code.visualstudio.com/>
- Freely available text editors

Grading

The following percentages are tentative and may be changed at any time:

Attendance, participation, professionalism	5%*
Preparation for Class (Examples, Comprehension Checks, Checkpoints)	10%*
Problem Set Assignments	30% **
Programming Projects	20%
Written Quizzes	20% **
Final Exam	15%

* *Participation/submission of 85% or more is necessary for full credit. Late class preparation submissions will not be accepted.*

The lowest two problem set scores and one lowest quiz score will be dropped. The programming project assignments are required and will not be dropped under any circumstances.

There are no makeup quizzes or extensions on homework – missed quizzes and assignments can be dropped as one of the low grades in each category.

Final letter grade cutoffs will be determined at the end of the course. Grades are not curved; however, we will take into account your overall effort for the entire semester, but this can only help your grade, not hurt it.

Policies and Miscellaneous

Assignments are due on the date stated on the homework assignment (to be posted on web).

- Assignments received within 0-24 hours of the deadline will be accepted with a 10% penalty.
- Assignments received more than 24 hours past the deadline will not be accepted or graded.

Plan your work accordingly, and work on all assignments as soon as they are given so you can ask questions in class and get assistance in the labs and tutoring hours.

Students are responsible for ensuring that assignments are correctly submitted. If you have a question or problem, seek help from CS 124 staff immediately.

Attempts to submit homework by email without prior permission will be ignored. Technical problems (computer crashes) will not be cause for an extension. You must backup your work.

The official administrative business of this class will be conducted via Piazza.

All class correspondence should take place on Piazza.

Grade questions/disputes, explanation of absence, etc. will be processed via email or Piazza so that we both have a written record of what was agreed.

No special make-up work will be accepted after the end of the semester.

In the event of a documented major medical problem, an Incomplete may be requested pending the submission of complete work. However, make up work “to improve one’s grade” will not be accepted.

Grades are not negotiable. Don’t even ask – just do the work and you’ll get the grade you deserve. Of course, please bring any clerical grading errors to our attention and we will gladly fix them.

Time Commitment. This is a 4-credit class. Each class meeting will require several hours of preparation and additional work outside of class. You should expect to devote 15-20 hours per week to this course at a minimum. The workload will vary week-to-week. A minimum of 180 hours of engaged academic time is expected.

Expected minimum workload per week:

Preparation for Class:	2-3 hours
Class Lectures/Discussions:	4 hours
Problem Sets:	7-12 hours
Review/Studying:	2

Students with Disabilities. Clark University is committed to providing students with documented disabilities equal access to all university programs and facilities. Students are encouraged to register with Student Accessibility Services (SAS) to explore and access accommodations that may support their success in their coursework. SAS is located on the second floor of the Shaich Family Alumni and Student Engagement Center (ASEC). Please contact SAS at accessibilityservices@clarku.edu with questions or to initiate the registration process. For additional information, please visit the SAS website at: <https://www.clarku.edu/offices/student-accessibility-services/>

Welcoming Class Atmosphere. We strive to make my classroom welcoming and productive for all students. Please let us know what name or pronoun you prefer to use. Please bring to our attention any issues or concerns you have about the classroom atmosphere.

Notice to students: Faculty Members are “Responsible Employees”

In compliance with Federal law, the Faculty member(s) and Teaching Assistant(s) associated with this course are considered “Responsible Employees” by Clark University. As such, they are required to report all alleged sexual offenses to the University’s Title IX office (titleix@clarku.edu). The only exceptions to this reporting responsibility are the community members who have been designated and/or trained as “Confidential” Sources. This includes the professional staff in Clark’s Center for Counseling and Personal Growth (508-793-7678) and the medical providers at the Health Center, (508-793-7467) In addition, Professor James Cordova (jvc.confidential@clarku.edu) Professor Kathy Palm Reed (kpr.confidential@clarku.edu) and Professor Andrew Stewart (als.confidential@clarku.edu) are also considered confidential resources.

Withdrawing from the Course

If you feel that you want to drop or withdraw from the class, please schedule a time to talk to us about it as early as possible; we want to help you succeed, but you need to ask for help.

Course Flexibility The instructor reserves the right to adjust assignments, the organization of the course, and other class elements as needed. Material changes to the course or policies will be communicated via the course Piazza site.

Plagiarism, Collaboration, and Collusion

Unless otherwise specified, CS124 homework assignments are specified as individual **independent work** or as pair/group programming.

It is the student's responsibility to know and understand the Clark University Academic Integrity policy.

In addition to the definition of plagiarism in the handbook, **with respect to CSCI124, plagiarism is specifically defined to include (but is not limited to) the following:**

- collaboration on the solutions/code you write
- copying any part of someone else's assignment/program, even if you have permission and/or have modified the code
- sharing or giving your assignment/code or even a subset of your assignment/code to another student to review
- reviewing another student's solution (including from past semesters)
- reviewing solutions on the internet
- Using AI tools to write, edit, outline, or improve code related to any part of an assignment

It is my policy to use automatic plagiarism detection software, and suspicious similarities will be uncovered. The University takes acts of cheating and plagiarism very seriously; violators may be suspended or fail the course.

What is acceptable cooperation?

For individual-only and pair/group-optional problems, you may discuss ideas and approaches with others (provided that you acknowledge this in your solution), but such discussions should be kept at a high level, and should not involve actual details of the code or of other types of answers.

Cooperation is recommended in understanding programming concepts and system features. You are encouraged to discuss the labs, the homework problem statements and expected output, and to seek and receive help with Python, development tools and other software.

However, each student must write his or her own solution/code and other deliverables independently.

For working on assignment where **pair/group programming** is allowed:

- You may not split up the work and complete it separately.
- You must work together at the same computer (or remote screen sharing) for every problem that you complete as a pair/group. While you are working, the screen should be visible to each partner. One person should type, while the other person observes, critiques, and plans what to do next. You must switch roles periodically, and your solution should be a true collaborative effort. Each partner must be able to explain the entire submission.
- You must both submit the same solution to each problem that you did as a pair, and you must clearly indicate that you worked on the problem as a pair by putting your partner's name at the top of the file.