CS 152
Computer Programming Fundamentals

Brooke Chenoweth

University of New Mexico

Spring 2025

Contact Info

Instructor:	Brooke Chenoweth
Email:	bchenoweth@cs.unm.edu
Office:	Room 2060 in Farris Engineering Center
Web site:	`cs.unm.edu/~bchenoweth/cs152`

Schedule — Lectures (required)

- 10:00 am - 10:50 am
- MWF
- Centennial Engineering Center 1041

Schedule — Labs (also required)

Lab	Time	Day	Location
001	11:00 am - 11:50 am	M	Cent Eng Center 1030
002	11:00 am - 11:50 am	W	Mech Eng 210
003	11:00 am - 11:50 am	F	Cent Eng Center 1028
004	12:00 pm - 12:50 pm	M	Cent Eng Center 1032
005	12:00 pm - 12:50 pm	W	Cent Eng Center 1030
006	12:00 pm - 12:50 pm	F	Cent Eng Center 1032
007	9:00 am - 9:50 am	W	Cent Eng Center 1028
008	9:00 am - 9:50 am	F	Cent Eng Center 1028
009	11:00 am - 11:50 am	F	Cent Eng Center 1032

Each lab session will have some sort of in lab exercise as well as time for questions and course help.

Lab 009 is new!

- We had so many people on the waitlist that we made an additional section.
- There is room in a couple sections now.
- There are still people on the waitlist.
- If you would like to switch to an another section, please do so this week.
    - If you've done work on Canvas and then switch sections, you might have to resubmit.

Mech 220 flooding!

There was a burst pipe in Mechanical Engineering that leaked into room 220.
The affected labs have moved to new locations.

- Lab 001 (M 11:00am-11:50am) will be in CENT 1030
- Lab 003 (F 11:00am-11:50am) will be in CENT 1028
- Lab 005 (W 12:00pm-12:50pm) will be in CENT 1030

Office Hours

- Office Hours: TBA
  I've posted a survey on Canvas to find the classes preferences for days/times, remote vs in person, before choosing my office hours. Hours will be posted on the course website once determined.
- You may attend regular office hours without an advance appointment. If you want to meet at another time, make an appointment by email.
- Section leaders have office hours, too! (TBA, Check course website)
- Feel free to ask any of the section leaders for help.

Grading

- 40% Programming Assignments
- 40% Exams (midterm and final)
- 10% Canvas quizzes, surveys, etc.
- 10% Lab exercises and participation
    - Lab section exercises are in person, but sometimes may also have a component to submit on Canvas.

Assignments and Projects

- Assignments must be submitted in Canvas to receive credit.
  - Contact Canvas support if you are having technical difficulties.
- It is your responsibility to make sure you submit the correct file.
- Don't wait until the last minute to submit.
- Submit early, submit often!
  - We'll grade your most recent submission.

Extension Days

- Ideally, you'll never need to turn in an assignment late.
- However, life happens!
- You have 10 extension days to spend through the term.
- Max 3 days per programming assignment.
- Use them wisely.
- You don't need to ask before using regular extension days.
- Contact me if these will not be enough, preferably *before* you are late.

ARC Accomodations

- The Accessibility Resource Center provides accomodations with students with disabilities.
- For example: Extra time and/or quiet location for exams
- `http://arc.unm.edu`
- Please take advantage of their services if applicable

Canvas

- `http://canvas.unm.edu`
- Assignment submissions
- Discussion forum
- Surveys and quizzes
  - Office hour survey, welcome discussion, and syllabus quiz are there now!

Working Together

- Working together and helping one another on all projects is highly encouraged. This includes discussion of:
  - project specification
  - algorithms
  - data structures
  - test cases
  - **Not code!**
- Do *not* share code.
- It is considered cheating to leave your code (paper or electronic copies) where others can find it. You responsible for the security of your intellectual property.

Cheating

- Don't cheat.
- Using books, websites, other people as resources is expected, but document it.
- If unsure, talk to us first.
- Understand your code! If you didn't actually write it, you likely don't understand it.
- Trying to "help" a friend by sharing your solution is also cheating.

Is this cheating?

- You ask a friend for help with your program.
- They give you their solution from when they took the course.
- You copy their code into your project.

Is this cheating?

- You ask a friend for help with your program.
- They give you their solution from when they took the course.
- You copy their code into your project.

You didn't write that code!

Is this cheating?

- You ask a ChatGPT for help with your program.
- It generates a solution.
- You copy its code into your project.

Is this cheating?

- You ask a ChatGPT for help with your program.
- It generates a solution.
- You copy its code into your project.

You didn’t write that code!

Is this cheating?

- You ask a friend for help with your program.
- They describe a general approach, draw a helpful diagram, and suggest a Java library they like to use.
- You discuss the approach and ask for clarification.
- You read the documentation for the library and decide if it would help.
- You think a while and code up your own solution.

Is this cheating?

- You ask a friend for help with your program.
- They describe a general approach, draw a helpful diagram, and suggest a Java library they like to use.
- You discuss the approach and ask for clarification.
- You read the documentation for the library and decide if it would help.
- You think a while and code up your own solution.

You *did* write that code!

Topics

- Variables
- Basic program flow
- Conditional branching (if statements)
- Loops (for, while)
- Arrays
- Input/Output
- Objects and classes
- Basic graphics

Summary

- Go to class and labs
- Keep up with the websites
- Expect some sort of work each week
- Be proactive!
- Form study groups
- Ask questions
- The instructors are there to help you

To do

- Visit course website
  - Slides will be posted after the lecture.
- Visit Canvas site
  - Take welcome quiz and office hours survey
  - Visit discussion forum, introduce yourself
- Contact ARC if you might need it

Computing

- In the computer, it is all just numbers.
- A computer file is just a sequence of 1s and 0s.
- Computers do simple things
    - Set the intensity of the red, green and blue light given off by a particular pixel on a display.
    - Send a precisely timed sequence of hi and low voltage values to a hard disk controller, USB port, or wireless transmitter.
    - Add, Subtract, Multiply or Divide two numbers.
    - Read or Set the voltage state of a particular memory circuit.

Complexity from Simplicity

- The complex things we see computers do are the results of programs.
- Even “short” programs are huge and complex lists of simple computer instructions.

Programming Language

- A *programming language* is a set of symbols and rules designed for humans to more easily represent computer instructions.
- In this class, the programming language we will be using is Java.

Computer Programs and Recipes

- Like a recipe, say for cooking quiche, a computer program is a sequence of steps.
- In a recipe, conditional logic, if it exists at all, is trivial. For example:
    1. Bake for 50 minutes.
    2. Remove from oven.
    3. Insert a knife into the center and remove. If bits of egg cling to the knife, then return quiche to oven and bake for another 10 minutes.
    4. Repeat this until the knife comes out clean.
- In a computer program, it is common for there to be 100s or 1000s of conditions with complex nestings and other interrelations.

Programming vs Natural Language

- The entire Java vocabulary consists about 50 reserved words.
- There are many Java *classes*, such as `Math` and `Image`. However, these are the *proper nouns* of the language.
- A person can be fluent in a language without knowing the vast majority of its proper nouns.
- Proper nouns are learned as needed, and can be forgotten when no longer needed.
- Like natural languages, programming languages have punctuation and syntax rules (e.g. In Java, every statement is ended with a semicolon). Programming languages, however, have fewer rules than natural languages.

Small Language with Complex Usage

- Programming Languages are much smaller and easier to learn than natural languages.
- However, programming languages are primarily used to express complex branchings of conditional logic that far surpass common uses of natural languages.
- Logic skills have strong carryover from one programming language to another.