CS 351
Design of Large Programs

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/cs351`

Schedule

- Lectures (required)
  - 9:30 am - 10:45 am TR
  - Dane Smith Hall 136
- Labs (also required)
  - Lab 001: 11:00 am - 11:50 am T
  - Lab 002: 11:00 am - 11:50 am R
  - Centennial Engineering Center B146A

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 or in person.
- TA also has office hours (TBA, check course website)

Grading

- 85% Projects
  - 5 projects
  - Initial projects: sequential, individual
  - Later projects: concurrent, groups
- 15% Lecture, lab exercises, participation, etc.

Technology

- Programming language: Java
  - We will be using JDK 21, in particular the Azul Zulu build that includes JavaFX
  - 21 is a LTS version and it's good for all of us to be on the same one.
- GUI library: JavaFX
- IDE: IntelliJ
- Version control: Git
- Project hosting: Github classrooms
- Project grading/testing: CS Linux machines

Computer Access

- Your projects need to work on a CS Linux machine.
- Get a CS account (in addition to your UNM account)
- To work remotely: Use SSH to connect:
  - **moons.cs.unm.edu**
  - **trucks.cs.unm.edu**
  - B146 machines
- With a CS computer account you can access ***.cs.unm.edu** and use the CS Linux lab in Farris as well as the CENT B146 lab.

How to Get a CS Account

Email cssupport@cs.unm.edu from your UNM email account. Include:

- Your full name
- Your UNM ID number
- The CS course(s) you are taking
- A picture of your Lobo ID attached to the email

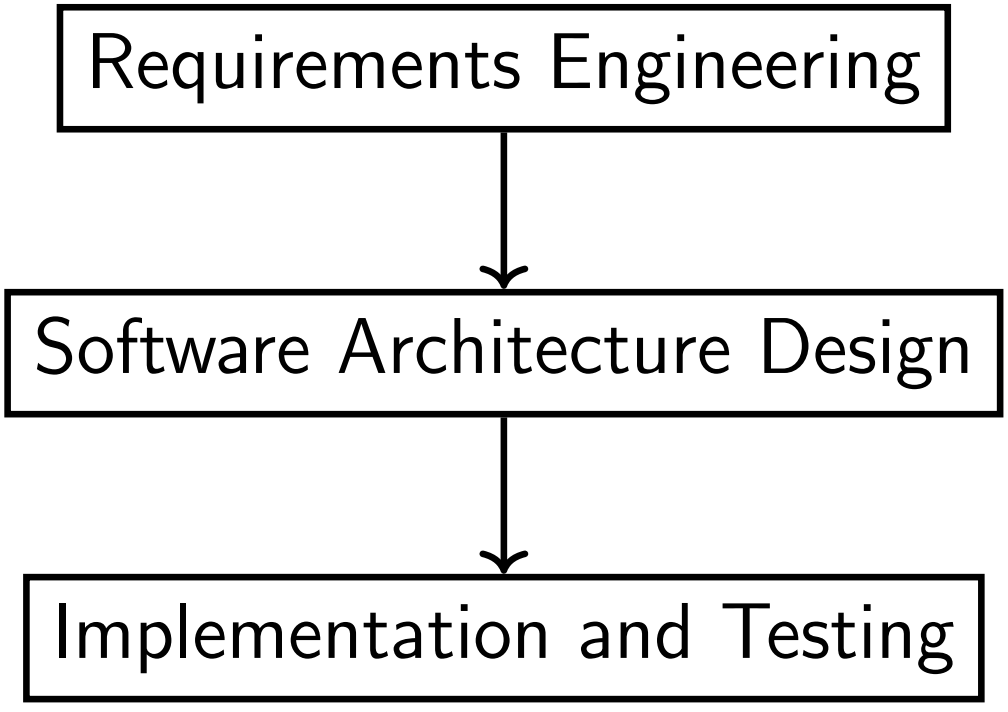
Project Submission

- Projects will be hosted on Github classroom from an assignment link posted on Canvas.
- Follow the submission guidelines and coding standards posted on course website.
- Don't wait until the last minute to get started
- We'll see whatever your latest version of the project is when we look at it, so just make sure you push your work regularly to Github.

Prerequisite Skills

- Functions and Procedures
- Recursion
- Classes and Objects

Software Development Lifecycle



CS460

CS351, CS460

CS251, CS351, CS460

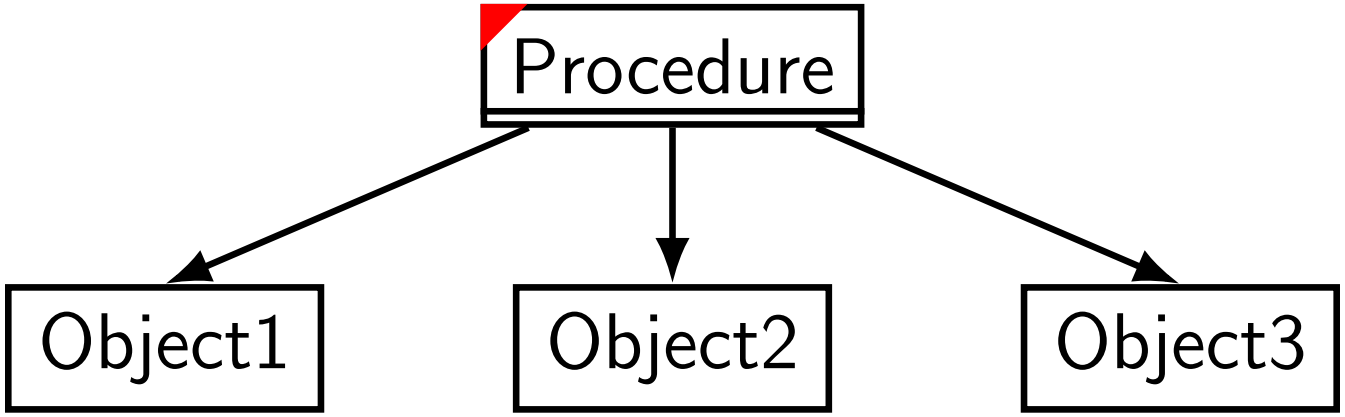
Course Outline

- Intro
  - Object Oriented Design
  - Object Oriented Programming
- Sequential Programming
  - Abstract data types
  - Classes, inheritance, interfaces, specification, notation
  - Complex data structures
  - Design patterns
- Concurrent Programming
  - Concurrency
  - Threads and synchronization
- Distributed Computing
  - Client-server model
  - Socket programming

Object Oriented Design

- A design paradigm that emphasizes:
  - Data and device encapsulation
  - Information hiding
  - Top-down hierarchical structuring
- The prototypical structure entails:
  - One main procedure
  - Several subordinate objects
- Highly complex system designs employ the same basic principles
- Object-oriented design can be employed even when the underlying programming language is not object-oriented

Object Oriented Design Pattern



Object Oriented Programming

- The concepts of *object* and *class* are explicit programming constructs in the language.
  - Objects: instantiated from class definitions
  - Classes: have associated code that is executed on behalf of instantiated objects
  - Classes are defined in terms of other classes by using inheritance
- Object-oriented programming languages simplify the implementation of object-oriented designs.
- A given design may have many different and distinct program representations.
- *Use of object-oriented programming languages does not guarantee clean design and proper encapsulation.*