CS 351
Design of Large Programs

August 23, 2021

Contact Info

Instructor: Joseph Haugh

Email: jhaugh@cs.unm.edu

Schedule

- Lectures
  - 9:30 am - 10:45 am TR Mitchell Hall 102
- Labs
  - Lab 003: 11:00 am - 11:50 TR CENT B146A
  - Lab 004: 12:00 pm - 12:50 TR CENT B146A

Grading

- 90% Projects
  - 4-5 projects
  - Initial projects: sequential, individual
  - Later projects: concurrent, groups
- 10% Lab exercises and participation

Technology

- Programming language: Java
  - We will be using Azul JDK 16 (see javafx pdf for more details)
- GUI library: JavaFX
- IDE: IntelliJ
- Version control: Git
- Project hosting: GitLab server at `lobogit.unm.edu`

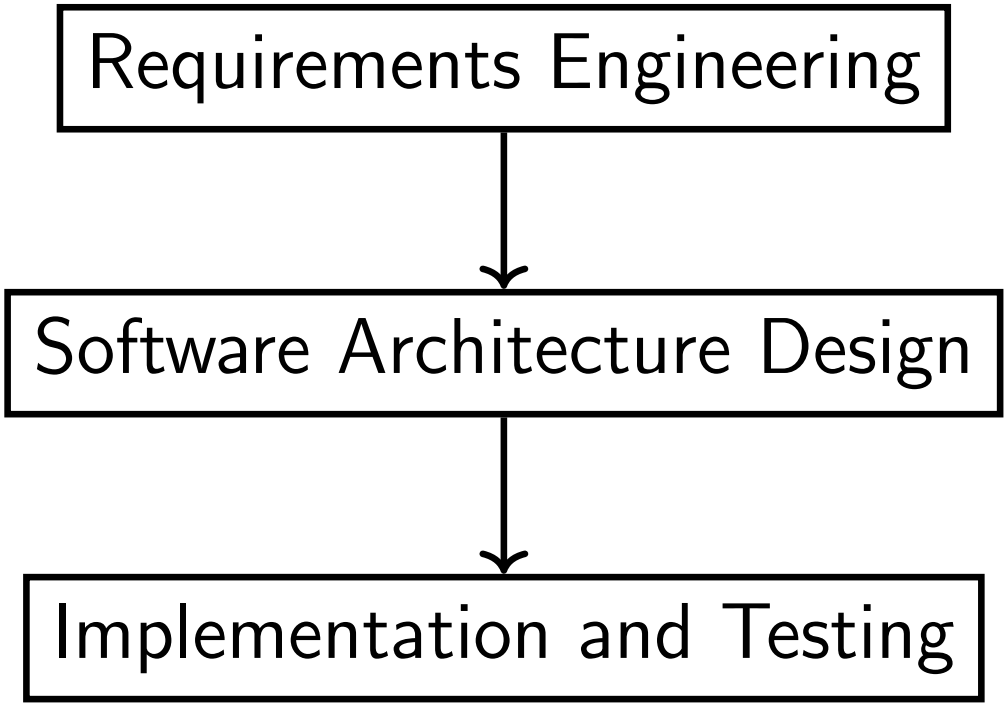
Project Submission

- Projects will be hosted on Lobo Git with a link submitted via UNM Learn.
- It is your responsibility to make sure you submit the correct link with the correct permissions on the project.
- Follow the submission guidelines and coding standards posted on course website.
- Don't wait until the last minute to submit.

Prerequisite Skills

- Functions and Procedures
- Recursion
- Classes and Objects

Software Development Lifecycle



CS460

CS351, CS460

CS251, CS351, CS460

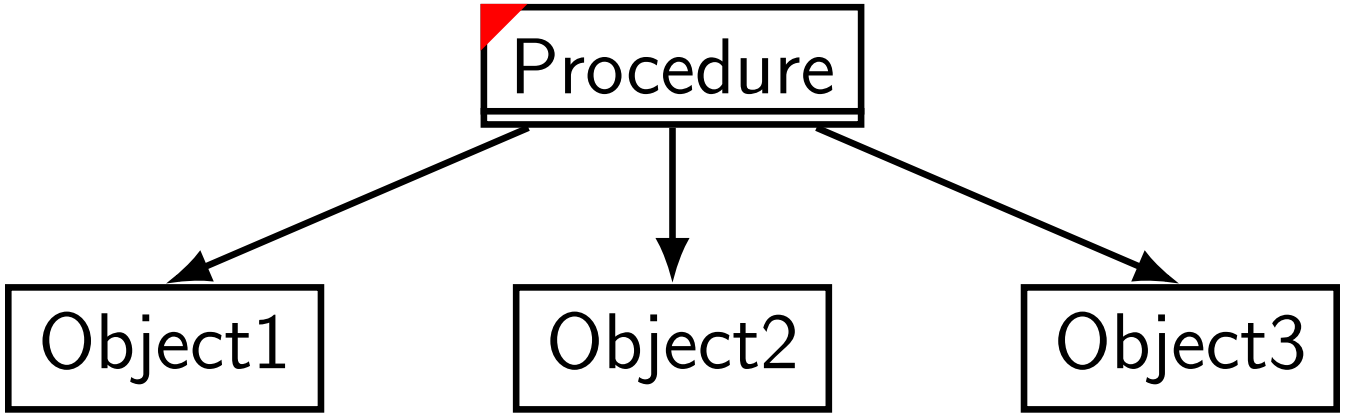
Course Outline

- 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.*