



Introduction to Software Engineering (CAS 313, Mondays and Wednesdays 2:30-4:15pm, Spring 2023)

Tentative Syllabus

The following is a *tentative* syllabus for EC327. The actual material covered will be depend on class intangibles, such as average student background and progress, and may follow a different order.

Introduction

- Operating System
- High-level programming
- Low-level organization

Abstractions

- Low-level
- Mid-level
- Higher-level

Infrastructure

- Testing and design
- Frameworks
- Data structures and algorithms

Tools

- Program development
- Version control

Introduction

Operating System

- *Linux*
 - File system
 - Command line
 - Text editing and processing
- *Windows, Android*

High-level programming

- *Basic C/C++*
- *Design, preprocessing, compilation, linking*

Low-level organization

- *bit/byte arithmetic*
- *basic logic*

Abstractions

Low-level

- *Hardware*
- *Computer architecture*
 - memory hardware
 - CPU, ALU
 - caching
 - bus
- *Machine language*
 - microcode
- *Assembly*
 - registers
 - instructions

Mid-level

- *Memory abstractions*
 - allocation
 - stack
 - heap
- *Multiprocessing*
 - processes
 - threads
- *Storage*
 - hard disks
 - page cache
- *System C*
 - Data primitives, constants
 - Selection
 - Loops
 - Functions
 - *the call stack*
 - *scope*

- *parameter calls*
- pointers
 - static and dynamic memory allocation
- **Arrays and C-strings**
- **file I/O**
 - *command-line arguments*

Higher-level

- *Object-oriented programming*
 - Packages and classes
 - Inheritance
 - design modifiers
 - constructors/destructors
 - Overloading, overriding
 - operators
 - **this** pointer
 - Polymorphism
 - virtual functions
 - dynamic/static typing
- *Language features*
 - Exceptions
 - Checked/unchecked
 - Propagation
 - Templates and generics
 - Standard Template Library
 - Containers
 - Algorithms
 - Left/Right values
 - Streams
 - Sockets
 - Threads and processes
 - communication
 - sandboxing
- *Mechanisms*
 - (Graphical) User Interfaces
 - Networking
 - Databases

Infrastructure

Testing and design

- *Unit and system tests*
- *Mocking and reliability*
- *Patterns and anti-patterns*

Frameworks

- *Android Development Kit*
- *Web Design*

Data structures and algorithms

- *Asymptotic analysis*
- *Data structures*
 - Linked lists
 - Stacks
 - Queues
 - Hash tables
 - Trees
- *Sorting*

Tools

Program development

- *Complex compilation*
 - Makefiles
 - gradle, maven
- *Programming style*
 - Variable naming conventions
 - Commenting and automated documentation
- *Integrated Development Environments*
 - Libraries
 - Event processing
 - Graphical User Interfaces
- *Debugging*
 - gdb
- *Optimization*

Version control

- *git*
 - branches, rebasing
- *Continuous Integration / Continuous Development*

- Ticketing
- Peer review
- Automated testing