

EC327 - Introduction to Software Engineering - Spring 2025

Instructor: Ed Solovey, PHO 531 Email: esolovey@bu.edu

- On-demand: <https://calendly.com/esolovey-bu/30min>

Course Objectives:

This course aims to provide a practical, bottom-up introduction to programming techniques, data structures, software design and engineering principles through the use of modern applications.

We will begin with an brief explanation of the hardware that powers modern computers, followed by an understanding of machine language and assembly language that controls the hardware and the operating systems that organizes that control. We will then proceed through fundamental elements of procedural programming languages, using system C as the case example, and continue with the principles of modern object-oriented programming, as embodied in C++ and its daughter languages (e.g., Java, JavaScript, Rust, C#, ...). The course will conclude with an introduction to elementary data structures and algorithmic analysis.

Throughout, we will develop core competencies in software engineering, including programming style, optimization, correctness, and program management and development lifecycle. We will also apply our work through modern applications from areas of security, graphical user interfaces, and networking, utilizing standard software engineering tools, such as Integrated Development Environments and large scale version control and ticketing systems.

Textbooks & Readings:

Cay Horstmann, Big C++ - Late Objects, Wiley, 2018

This book describes the fundamentals of C++, object-oriented programming, and data structures. It will serve as a good preliminary reference for our class, although much of the class material will go well beyond the book's depth. This book is required as readings from the book are assigned (in Blackboard) as prerequisites for lectures

Bjarne Stroustrup. The C++ Programming Language (4th Edition), Addison-Welsely Professional, 2013

The definitive guide to the C++ programming language by one of its original implementers. Written in very technical language, this work provides significant details about the language and

its use, although it is based on the 2011 version of the language (which is still in widespread use). This book is recommended

Assignments, announcements, course material, updated schedule, and other useful links

- posted on Blackboard (<http://learn.bu.edu>).

Course Outcomes:

As an outcome of completing this course, students will:

- Understand the six level computer architecture, how each level builds upon those below it, and how execution flows from the top of the pyramid down to its foundations.
- Become proficient in writing, testing, and debugging C/C++ code.
- Internalize foundational software engineering concepts such as data structures, scope, mutability, recursion, and object oriented principles and be able to apply them to languages other than C/C++.
- Understand the challenges of developing a non-trivial code base in a team environment and the techniques and tools that help address them.
- Establish foundations around reasoning about data structures, algorithm performance, and Big-O notation.

Topics:

1. Digital Logic
2. Six Level Computer Architecture
3. C foundations - function invocations and structs
4. C pointers & control flow
5. Memory Management
6. Recursion
7. Coding Best Practices
8. Object Oriented Programming
9. Polymorphism and Inheritance
10. Templates and Generics
11. Standard Template Library
12. Threads and Processes
13. Testing Philosophies and Unit Testing
14. Performance Analysis and Big-O
15. Data Structures
16. Sorting

Evaluation:

- **10%** - in class responses to interactive PollEverywhere questions
- **15%** - lab assignments
- **20%** - homework assignments
- **20%** - closed book midterm
- **15%** - team based final project (completed during second half of the class)
- **20%** - closed book final exam

Course Policies:

The most successful software engineers are also the most collaborative ones. No one innovates by themselves in a vacuum. We are not going to sacrifice collaboration in this class. However, it is important to evaluate your individual progress in this class. To that end, students are encouraged to collaborate with all of their classmates as much as possible, brainstorm ideas, help each other with set up, and getting around bugs. Additionally, certain assignments will encourage/require peer code reviews.

Students are allowed to consult LLMs for ideas, but never for copying and pasting code.

The lab and HW assignments that you submit must be your own code and words. If asked, you need to be able to explain every line of code that you submit.

The closed book exams will cover the same material as the class lectures and assignments, and ensure that students can reason through the assignments on their own will prepare them for the exams.

- **Academic integrity**
 - The homework and lab assignments must be the result of your individual work. You may discuss the content, general approach, and challenges of the assignments with your classmates, but not the detailed solution. You are expected to formulate your approach and write the solutions of HW problems by yourself. Copying the solution and/or answer from another student or source is considered cheating. **You may not submit ANY code not written by you.**
 - **You may not collaborate in any way on the exams.**
 - Clearly reference any sources you used in your work: books, internet, and your collaborators! This includes websites (e.g. Stack Overflow) and AI assistants (e.g. ChatGPT) in which case you are required to include details of the prompts that you use. **Note that copying code or answers from such sources, or from another student is considered plagiarism.**
 - Boston University's academic code of conduct, <https://www.bu.edu/academics/policies/academic-conduct-code/> will be strictly applied.
- **Copyright**

- All class material is copyrighted, and may not be shared publicly online by any means. This includes your own solutions to assignments.

Inclusion:

I consider this classroom to be a safe and supportive place where you will be treated with respect, and I welcome individuals of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability – and other visible and nonvisible differences. All members of this class are expected to contribute to a respectful, welcoming and inclusive environment for every other member of the class.

Accommodations for Students with Documented Disabilities:

If you believe you might have a disability that requires accommodations, requests for accommodations must be made in a timely fashion to Disability & Access Services, 25 Buick St, Suite 300, Boston, MA 02215; 617-353-3658 (Voice/TTY). See established policies and procedures: <http://www.bu.edu/disability/accommodations/>