

COURSE INFORMATION

EC327 – Introduction to Software Engineering Fall 2020

Location and Time

Lectures:

Monday and Wednesday, 2:30pm-4:15pm

All lectures will be held online: [Zoom link](#) (password: 3272020)

Labs (starting on 9/8/2020 Tuesday):

Monday 4:30pm-6:15pm

Tuesday 1:30pm-3:15pm

Thursday 1:30pm-3:15pm

Friday 10:10am-11:55am

- **You must register for ONE of these sections.** Please attend your section only. Lab sessions start and end at the given times sharply.
- **Monday lab session on 9/7:** Please attend one of the Tue, Thur, Fri sessions between 9/8-9/11 OR complete the tutorials on your own time watching the recordings.
- Lab sessions will be held **online** – please see Zoom link on Blackboard -> Information.
 - You may participate in the lab sessions in PHO305 if you desire or you may participate fully online. **You must follow BU's requirements for wearing a mask, maintaining social distancing measures, and others at all times if you choose to go to PHO305.** As the capacity of PHO305 is limited to 15 people, there is a [schedule](#) on Blackboard -> Information showing which weeks you can attend the lab in PHO305. You may request access to PHO305 on Zaius: <http://www.bu.edu/dbin/eng/zaius/>.

Staff

Instructor

Ayse Coskun (acoskun@bu.edu)

Office hours (starting on 9/8/2020 Tuesday):

Tuesdays 11am-noon and by appointment.

- To ensure faster scheduling for appointments, please email your availability for the *next 3 weekdays*.

GTFs

Pouya Haghi (haghi@bu.edu)

Hammas Saeed (hammas@bu.edu)

TAs

Ben Chan (chanben@bu.edu)

Prithika Ganesh (pvganesh@bu.edu)

Daniel Li (dli0793@bu.edu)

Panat Taranat (ptaranat@bu.edu)

Haoming Yi (yiha@bu.edu)

All GTF and TA office hours will be held online. Please see Blackboard for the Zoom link and the latest office hour schedule (Blackboard->Information).

Course Content

This course aims to introduce software design, programming techniques, data structures, and software engineering principles. The course is structured bottom up. We will begin with a brief explanation of the hardware that powers modern computers, followed by an introduction to machine languages that control the hardware and the assembly language that organizes that control. We will then proceed through fundamental elements of functional programming languages, using C++ as the case example, and continue with the principles of object-oriented programming, as embodied in C++ and its related languages Java, C#, and Objective C. We will demonstrate the use of Integrated Development Environments (IDEs) to design and manage large bodies of code. The course will conclude with an introduction to elementary data structures and algorithmic analysis. Throughout, we will introduce core competencies in software engineering, including programming style, optimization, debugging, compilation, program management, and dynamic memory allocations. We will also introduce some more advance concepts as time allows, such as hacking, graphical user interface programming, and basic networking. The course also includes a substantial project creating a *mobile application* using the Android Software Development Kit (SDK) or a *web app*.

Resources

Textbooks

- Y. Daniel Liang, *Introduction to Programming with C++*, Prentice Hall, 2014, 3rd edition.
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 beyond the book's depth. **(Required)**
MyProgrammingLab, which comes with the book, is not required; registering for it is up to you.

References

- Mark Allen Weiss, *Data Structures & Algorithm Analysis in C++ (3rd edition)*, Addison-Wesley, 2006:
This is a fairly easy-to-understand text on data structures in C++.
- Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms (3rd edition)*, MIT press, 2009:
This is the most complete reference for **data structures and algorithms** currently in use. It includes many advanced algorithms and data structures taught in subsequent courses.
- Lender Ammeraal, *C++ for programmers (3rd edition)*, John Wiley & Sons, 2000:
A textbook on C++ for those who are already comfortable with programming.
- Prata, *C++ Primer Plus (6th or 7th editions)*, Sams Publishing, 2005:
A thorough C++ reference.
- Dietel & Deitel, *C++ How to Program (8th or 9th editions)*, Prentice Hall, 2009:
A simplified but fairly complete reference for the C++ programming language.
- Bjarne Stroustrup, *The C++ Programming Language (3rd Edition)*, Addison-Wesley, 1997:
The author is the creator of C++. This is a definitive reference.
- Koenig and Moo, *Accelerated C++: Practical Programming by Example*, Addison-Wesley, 2000:
Based on an accelerated 2-week course taught at Stanford, provides a jump-start to key (and advanced) concepts in C++.
- Scott Meyers, *Effective C++: 55 Specific Ways to Improve Your Programs and Designs*, Addison-Wesley Professional, 2005:
Techniques for writing clear, correct, efficient C++ code, useful after gaining some expertise in C++.
- cplusplus.com – The C++ resources network: <http://www.cplusplus.com/>.

Course Elements

Blackboard

You are responsible for checking the Blackboard page for *EC327 – Fall 2020* regularly. Blackboard will contain handouts, assignments, lab material, practice exam information, and your grades as they become available. **You will NOT use Blackboard to submit homework assignments.**

Piazza

The class has a Piazza discussion board and you are encouraged to post questions and respond to other students' questions. Piazza will be monitored by the TAs and the instructor. Posting assignment solutions on the discussion board is strictly forbidden. Please make sure your comments on Piazza are respectful towards others and are related to the course.

Communication with the EC327 Staff

Please make sure to include **EC327** in the subject line of any email you are sending to the instructor or the TAs to receive timely responses. Before emailing, please do check the Blackboard announcements and the Piazza discussion board as your question may have been answered already.

Lab Sessions

Lab sections are offered four times each week (see above for schedule) and focus on the practical implementation of the topics covered in class AND specialized topics (e.g., UNIX/LINUX, Android, Java, etc.). You **MUST** attend the lab section you are registered for. There will be at least 6 lab assignments throughout the semester, which will be **PASS/FAIL** and must be turned in within the time frame provided (typically within 90 minutes once you start the lab assignment). Each week lab assignments cover new topics that will either be done **individually or in groups of two students**.

Quizzes (typically 5 total)

Quizzes will be given during class. Their main aim is to test basic understanding of the course material. Quizzes may also include questions from labs or homework assignments. Students in remote time zones can start taking a quiz within 24 hours of its post date/time. Once a quiz is initiated, it will have to be completed within a short amount of time (e.g., typically within 10-15 minutes).

Homework Assignments (8 total, tentatively)

All homework assignments **must be completed individually**, although you may discuss *general* suggestions and questions with others in the class. Homework assignments will, in most cases, will focus on C++ programming. Programming assignments have been created for you to build your mastery of the core programming concepts and material, and they are at the heart of the course.

Any written code or answers that you submit must be completely your own work! You may not copy any code from anyone else, and you must **never** look at anyone else's code when working on your homework assignments. **This class uses automated software to detect similarities among submitted assignments.**

Unless otherwise stated, you may use any development environment you wish, as long as it is ANSI C++ compatible. **Please make sure your code compiles and runs on the target environment (Linux, specifically eng-grid) before submitting. Submitted code that is not compiling will strictly receive a 0 grade.**

Late Penalties

- Lab assignments can only be submitted within 24 hours of the lab session you are in. **NO late labs.**
- Homework assignments may be submitted up to 3 days late at the cost of a **20% fixed penalty** (e.g., submitting a day late and 3 days late is equivalent). If you submit your assignment multiple times

(e.g., before and after the deadline), only the latest submission will be taken into consideration and late penalty, if any, will be applied accordingly. No points will be given to solutions submitted after the 3-day period following the deadline.

- Penalties may be removed **only** for legitimate excuses with written, dated documentation of illnesses.

Project

There will be a group project assignment. Details will be provided roughly a month before the end of the semester. The aim of this team project is to design and build a marketable Android app or a web app. You will be graded on the front-end design, back-end design, and the marketability of the application. You will be using Android Development Kits, C++, and Java (a tutorial will be provided).

Grades

All grades will be curved. This is NOT a precise process and is a function of class average, improvement, class participation, and providing a balanced distribution of letter grades. The final grades will depend on our assessment of the class as a whole. Raw scores will be computed based on the following weights:

- Quizzes (3 best scores out of 5 quizzes) (15%)
- Lab assignments (5 best scores out of at least 6 lab assignments) (10%)
- Homework assignments (6 best scores out of 8 assignments) (45%)
- Project (15%)
- Exam (15%) – Tentatively during the final exam week, TBD. Accommodations will be made for students in remote time zones. Practice exams will be given in advance.

Collaboration Policy

All students are responsible for reading the Boston University academic conduct policy. Dishonesty in representing one's academic work (cheating or plagiarism) is a serious ethical violation and will be reported according to BU policy. You may use any textbooks or web sources (not run by a class member) when completing your assignments or labs (but not quizzes or exams) subject to the following strict conditions:

1. You must clearly acknowledge and cite all your sources (e.g., stack overflow).
2. You must write all answers in your own words. **All code must be your own.**
3. You must be able to fully explain your answers upon demand.

You may collaborate with people as follows, **unless explicitly stated otherwise in writing by the instructor**:

- Quizzes and exams: NO ONE
- Homework assignments: General ideas – anyone. Specific work and implementation – NO ONE.
- Lab assignments: Assignment-specific; could be done individually or in teams.

A good rule of thumb is that discussions on Zoom or in front of a white board are generally okay while discussions in front of computers or via shared screens looking at code are potentially dangerous. **If you are unsure about an activity, please ask the instructor first.**

BU Covid-19 Policies

This course is mostly going to be delivered online. However, please note that masks are required and face coverings must be worn over the mouth and nose at all times when in public spaces on campus, including classrooms and labs (e.g., if you choose to attend the lab sessions in PHO305). You should be prepared to show proof that you are compliant with health attestations and testing when on public spaces in campus. All students are expected to follow all university guidelines with respect to daily symptom checks, testing, social distancing, and mask wearing. For a detailed description of official BU policies regarding COVID, please visit: <http://www.bu.edu/dos/policies/lifebook/covid-19-policies-for-students/>