COURSE INFORMATION
EC327 – Introduction to Software Engineering
Fall 2021
Instructor: Prof. Douglas Densmore (dougd@bu.edu)

Location and Time

Lectures: Monday and Wednesday, 2:30-4:15pm, PHO 211

Labs:
Monday 4:30pm-6:15pm, PHO 307
Tuesday 1:30pm-3:15pm, PHO 307
Thursday 1:30pm-3:15pm, PHO 307
Friday 10:10am-11:55pm, PHO 307
(^ Seats are reserved for people in the class; open seats are on a first come, first serve basis)

Staff

Instructors

Douglas Densmore (dougd@bu.edu, 617-358-6238, **610 Commonwealth Rm 403**)
Office hours: Tuesday noon to 1pm (https://bostonu.zoom.us/j/4602829913), also by appointment. *
Tuesday Lab+

Graduate Teaching Fellows
Pujan Paudel (ppaudel@bu.edu); Office Hours: **Fridays 2:30pm to 4:30pm** (and by appt.); **Monday Lab**+

Undergraduate Teaching Fellows
Keven Deoliveira (kevend@bu.edu); Office Hours: **Tuesday and Thursday 2-3pm** (and by appt.); **Friday Lab**+

MS Teaching Fellow
Nyrika Bhargavaram Renuka (nyrikabr@bu.edu); Office Hours: **Monday 7-9 pm** (and by appt.); **Thursday Lab**+

+ Lab leader
* Please see Blackboard for potential updates.

Course Content

This course aims to introduce software design, object-oriented 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 an Integrated Development Environment (IDE) such as Microsoft's Visual Studio to design and manage large bodies of code and versioning systems such as Git (and development platforms like GitHub). 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 allocation. The course also includes a substantial project which may include creating a mobile application using the Android Software Development Kit (SDK).

Resources

Textbooks

- Behrouz A. Forouzan and Richard Gilberg, C++ Programming: An Object-Oriented Approach, 1st Edition, McGraw Hill, 2020.
  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. **First semester using this book!** **(Required)**

References

(Core Additional Textbooks)

- Y. Daniel Liang, *Introduction to Programming with C++*, Prentice Hall, 2014, 3rd edition.
  This is the old course textbook. We are going to try a new book this semester that I think has better examples and is organized better.
- Bjarne Stroustrup, *The C++ Programming Language (4th Edition)*, Addison-Wesley, 2013:
  The author is the creator of C++. **This is a definitive reference. (Recommended as a long-term resource; highly recommended)**

(Reference Textbooks)

- Lender Ammeraal, *C++ for programmers (3rd edition)*, John Wiley & Sons, 2000:
  This is a textbook for the C++ programmer who is already very comfortable with programming.
- Prata, *C++ Primer Plus (6th or 7th editions)*, Sams Publishing, 2005:
  This is a fairly thorough C++ reference.
- Dietel and Deitel, *C++ How to Program (8th or 9th editions)*, Prentice Hall, 2009:
  This is a simplified but fairly complete reference for the C++ programming language. This is good introductory textbook.

(Computer Science Topics)

- 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. **(Recommended as a long-term resource; likely used in EC330)**

(Web Resources)

- cplusplus.com – The C++ resources network: http://www.cplusplus.com/.
- stackoverflow.com – Lots of programming examples, resources, community, etc.

Course Elements

Blackboard:

You are responsible for checking the Blackboard page for *EC327 – Fall 2021* **regularly**. Blackboard will contain handouts, homework, programs, lab material, practice exam information, and your grades as they become available. **Homework (HW) and Program (PA) submission mechanisms are TBD**. Please check Blackboard for instructions for each homework and programming assignment.

Piazza:

You can post your questions on the Piazza discussion forums. This is an excellent way to get help very quickly and benefit from both instructor and fellow classmate resources. Information on how to sign up for Piazza will be posted as a Blackboard announcement. **Please check Piazza before sending staff email.**
http://piazza.com/bu/fall2021/engec327/home

Lab Assignments -- Lab location: PHO 307 (UNIX)

Lab sections are offered four times each week (see the beginning of the syllabus for the 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 unless explicitly permitted otherwise.** For example, if you registered for the Monday lab section, then you MUST attend the Monday lab section the whole semester. In the first half of the semester, lab assignments will be handed out, which are **PASS/FAIL** and must be turned in at the end of every lab section. Each week lab assignments cover new topics that will either be done **individually or in groups (of two students)**, depending on the covered topic. In the second half of the semester (i.e. after the Midterm week), the lab sections will be dedicated to implementing Programming Assignments 3 and 4. Those lab sections are also **PASS/FAIL** depending on each student's participation and collaboration.

Homework Assignments (HW) (4 total)
All homework assignments **must be completed individually**, although you may discuss *general* suggestions and questions with others in the class. Homework assignments (with the exception of HW1 which is assembly programming) will focus on programming theory and concepts and will typically be written assignments involving short answer responses.

Programming Assignments (PAs) (4 total)
Programming assignments are the heart of the course. These have been created to allow you to show your mastery of the core C/C++ concepts and material. **Any written code or answers that you submit must be completely your own work!** You may not copy **any** code from anyone else in class, and thus you cannot use others' code in completing your homework assignments. **This class uses automated software to detect similarities among submitted assignments. Cheating will result in a 0% on the assignment and further disciplinary action.**

PAs will be released earlier than when the previous PA is due. This is to provide students with the maximum amount of time to do the assignments. Each PA will have a due date but this allows those that finish PAs early to immediately start on the next assignment. Historically PAs at the end of the semester have been a source of stress. Hopefully this will alleviate this problem provided that students employ proper time management skills and work steadily over the course of the semester.

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) before submitting.** As a registered student in this course, you may get a free copy of Visual Studio.NET for your use in this class through our department's MSDN Academic Alliance, and you will get an e-mail describing how this may be done. If you have registered for this class sufficiently early, your BU ID should get you access to PHO 307, PHO 305, and PHO 115 during open lab hours. Otherwise, please submit your request through Zaius (http://www.bu.edu/dbin/eng/zaius/).

Late Penalties:

- Labs can only be done during the week they are offered. **NO late labs.**
- Programs, with the exception of PA4, may be submitted up to a week late at the cost of a **30% fixed penalty** (e.g., submitting a day late and a week late is equivalent). It is in your best interest to complete as many programming questions as possible before the deadline. If you have missing questions in your original submission, you may complete and submit the missing solutions during the following week. Any submissions after the deadline will be subject to the 30% penalty. No points will be given to solutions submitted after the 1-week period following the deadline.

- HW There will be a **10% penalty** per day for late homework, but up to a **maximum of two days late**.

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

Project:
There will a group project assignment. Details will be provided roughly a month (**November 10th according to the schedule**) before the end of the semester. The aim of this team project is to design and build a team software project commensurate in scope to the size of the team. An example project could be a marketable Android app. You will be graded on the front-end design, back-end design, and the marketability of the project. You can use other software language and environments (e.g., Android Development Kits, C++, and Java).

Quizzes:
There will be at least three quizzes given at random intervals during class**. They will NOT be announced and you can't ask when they will occur.** Their main aim is to test basic understanding of the course material and to promote regular lecture attendance. Quizzes may also include questions from labs, programs, and homework. **Quizzes will occur during class and cannot be taken if you are absent without prior approval.**

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 grade and which grade we assign to class average will depend on our assessment of the class as a whole.

Raw scores will be computed based on the following approximate weights:

- Quizzes (at least 3) (3%)
- Labs (7%)
- Homework (15%)
- Programs (30%)
- Project (15%)
- Midterm exam (15%) – **October 25th, 2021**. During class time. **Location(s) will be assigned on a per student basis.**
- Final exam (15%) – **TBD - See Blackboard for Updates**

Collaboration

All students are responsible for reading the university academic conduct policy. Dishonesty in representing one's academic work is a serious ethical violation and will be reported according to university policy.

Cheating and plagiarism will be taken **very seriously**. You may use any textbooks or web sources (not run by a class member) when completing your homework, programs, 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 and programs: General ideas – anyone. Specific work – NO ONE.
- Labs: Assignment specific

A good rule of thumb is that discussions on whiteboards and with pencil and paper are okay while discussions with computers, code, electronics, etc. are potentially dangerous. **When in doubt, ask!**

* Failure to meet any of the above conditions could constitute plagiarism and will be considered cheating in this class. **If you are unsure about an activity, please ask the instructor first.**