

## COURSE INFORMATION

### EC327 – Introduction to Software Engineering Spring 2022

---

#### Location and Time

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

#### Labs (starting on 2/1/2022 Tuesday):

Tuesday 3:30pm-5:15pm, PHO 307 (Instructors Pujan Paudel, Paige Deveau)

Wednesday 4:30pm-6:15pm, PHO 307 (Instructor Pranet Sharma)

Friday 2:30pm-4:15pm, PHO 307 (Instructor Manuel Segimon, Jin San Yoon)

You must register for ONE of these sections and attend your section only. Lab sessions start and end at the given times sharply.

#### Staff

##### Instructor

Gianluca Stringhini ([gian@bu.edu](mailto:gian@bu.edu), PHO 331)

**Office hours: Mondays 4:30-5:30pm and Wednesdays 4:30-5:30pm, also by appointment.\***

##### GTFs

Pujan Paudel ([ppaudel@bu.edu](mailto:ppaudel@bu.edu))

**Office hours: Fridays, 1-2pm, PHO 307 also by appointment.\***

##### TAs and Graders:

Paige Deveau ([pdeveau@bu.edu](mailto:pdeveau@bu.edu))

Manuel Segimon ([manuelsp@bu.edu](mailto:manuelsp@bu.edu))

Pranet Sharma ([pranetsh@bu.edu](mailto:pranetsh@bu.edu))

Jin San Yun ([jinsaaan@bu.edu](mailto:jinsaaan@bu.edu))

#### Office Hour Schedule (starting 1/31/2022):\*

	Monday	Tuesday	Wednesday	Thursday	Friday
1-2pm					Pujan
2-3pm					
3-4pm				Pranet	
4:30-5:30pm	Prof. Stringhini		Prof. Stringhini		
5-6pm					

**All GTF and TA office hours will be held in PHO 305 and 307.**

*\*Please see Piazza for potential updates.*

## 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 procedural 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. 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 advanced 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).

## Textbooks

- Behrouz A. Forouzan and Richard Gilberg, *C++ Programming: An Object-Oriented Approach*, McGraw Hill, 2020, 1<sup>st</sup> 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)**

## 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 (6<sup>th</sup> or 7<sup>th</sup> editions)*, Sams Publishing, 2005: A thorough C++ reference.
- Dietel & Deitel, *C++ How to Program (8<sup>th</sup> or 9<sup>th</sup> 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 – Spring 2022* 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**

You are responsible for checking the Piazza page for *EC327 – Spring 2022* regularly. We encourage discussion in this class, and we will use Piazza for that. The instructor and the TAs will post additional questions and small exercises on Piazza to allow students to better grasp key concepts covered in the lectures and help fix potential misconceptions early. Students are encouraged to participate in online discussions, such as asking and answering questions.

### **Github classroom**

We will use github classroom for the homework assignments. Github classroom will host homework descriptions, will allow you to commit your code, and will list your grades for each homework as soon as they are available.

### **Communication with the Instructor and TAs**

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 Piazza discussion board to see whether your question has been answered already.

### **Lab Assignments -- Lab location: PHO 307**

Lab sections are offered four times each week (see above for schedule, all in PHO307) 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 5-6 lab assignments (most likely more) throughout the semester, which will be **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.

### **Quizzes (at least 3)**

Quizzes will be given at random intervals during class. Their main aim is to test basic understanding of the course material. Quizzes may also include questions from labs or homework assignments. **Quizzes will occur during the lectures and cannot be taken later if you are absent.**

### **Homework Assignments (6-7 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 either focus on programming theory and concepts, involving short answer responses, or in most cases, they will focus on C++ programming. Programming assignments have been created for you to build your mastery of the core C++ 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 or PHO307 machines) before submitting.**

If you have registered for this class sufficiently early, your BU ID should get you access to PHO 307 during open lab hours. Otherwise, please submit your request through Zaius:

(<http://www.bu.edu/dbin/eng/zaius/>).

### **Late Penalties**

- Lab assignments can only be done before the end of the lab session you are in. If you did attend the lab but did not complete the assignment by the end of it, you will receive **50% credit**.
- 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). It is in your best interest to complete as many questions as possible before the deadline and submit. 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.

### **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. 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:

- Class participation (5%)
- Quizzes (at least 3) (5%)
- Lab assignments (10%)
- Homework assignments (35%)
- Project (15%)
- Midterm exam (15%) – Location and time TBD.
- Final exam (15%) – Location and time TBD.

### **Collaboration Policy**

All students are responsible for reading the Boston University academic conduct policy. Dishonesty in representing one's academic work is a serious ethical violation and will be reported according to BU 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 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 whiteboards or with pen and paper are generally okay while discussions in front of computers and code 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.**