 COURSE INFORMATION  
EC327 – Introduction to Software Engineering  
Fall 2014  
Instructor: Prof. Douglas Densmore (dougd@bu.edu)  

Location and Time  
Lectures: Tuesday and Thursday, 4-6pm, PHO 206  

Labs:  
Monday 7pm-9pm, PHO 307  
Tuesday, Wednesday, and Thursday 6pm-8pm, PHO 307  
(* You must go to your assigned section; labs end at 9pm or 8pm sharp)  

Staff  
Instructors  
Douglas Densmore (dougd@bu.edu, 358-6238, PHO 335)  
Office hours: Tuesdays and Thursdays noon-1pm, also by appointment. *  

Lab Instructors  
Ernst Oberortner (ernstl@bu.edu, 358-6338, PHO 404/428)  
Office hours: Wednesdays and Fridays 10am – 11am, also by appointment. *  

Undergraduate Teaching Fellows  
Luke Sorenson (lasoren@bu.edu); Office Hours: Wednesdays 3-5pm (and by appt.); Monday Lab *  
Annie Lane (aelane@bu.edu); Office Hours: Wednesdays 10-12pm (and by appt.); Tuesday Lab *  
Carlos Cheung (cheung.solrac@gmail.com); Office Hours: T, Th 11-12pm (and by appt.); Wednesday Lab *  
Timothy Chong (ctimothy@bu.edu); Office Hours: Mondays 3-5pm (and by appt.); Thursday Lab *  

Grader(s)  
TBD *  

*Please see Blackboard 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 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. 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).
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)*

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. *(Recommended)*
- 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. *(Recommended)*

Course Elements

Blackboard:
You are responsible for checking the Blackboard page for EC327 – Fall 2014 regularly. Blackboard will contain handouts, homework, programs, lab material, practice exam information, and your grades as they become available. **You will NOT use Blackboard to submit homework and programs.** You will be using another mechanism for this *(Please check Blackboard for instructions)*. You can however post your questions on the Blackboard discussion forum.

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

Lab sections are offered four times each week (M 7-9pm, T, W, and Th 6-8pm; all in PHO 307) 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. 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.
HW Assignments (4 total)
All homework assignments must be completed individually, although you may discuss general suggestions and questions with others in the class. Homework assignments will focus on programming theory and concepts and will typically be written assignments involving short answer responses.

Programming Assignments (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 you must thus never look at anyone else’s code in completing 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) 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 117 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 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 (October 28th according to the schedule) 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).

Quizzes:
There will be at least three quizzes 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, 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) (5%)
- Labs (10%)
- Homework (15%)
- Programs (25%)
- Project (15%)
- Midterm exam (15%) – October 21st, 2014. Location(s) and time TBD.
- Final exam (15%) – December 18th, 2014 – 3-5pm - 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.