EC327 Introduction to Software Engineering

Syllabus

Spring 2021

Contents

1 Administration 1
   1.1 Times/Places .................................................. 1
   1.2 Contact Information ...................................... 2

2 Course content 2

3 References: Books 2

4 References: Web 3

5 Online Resources 3

6 Grading and Assessment 3
   6.1 Collaboration policy ........................................ 3
   6.2 Grading Components ....................................... 3

7 Labs 4

8 Expectations of Achievement Levels 4
   8.1 A-level student ............................................. 4
   8.2 B-level student ............................................. 4
   8.3 C-level student ............................................. 5
   8.4 D or F level student ................................. 5

1 Administration

1.1 Times/Places

The lecture is Monday and Wednesday 2:30pm-4:15pm on Zoom.

The lecture room is officially HAR 105, but whenever the in-person count is 27 or less, we will meet in PHO 117.
All lectures will be recorded and posted.

1.2 Contact Information

Outside of class, the best way to contact Dr. Carruthers is via the piazza discussion board.

Here are some alternatives to be used in emergency situations:

- email jbc@bu.edu
- phone/text message: 617-264-7939

The instructional team also includes:

TF Pouya Haghi haghi@bu.edu
TF Hammas Saeed hammas@bu.edu

2 Course content

This course aims to introduce software design, programming techniques, data structures, and software engineering principles. Computer hardware and organization. Machine and assembly languages. Fundamental elements of functional programming languages using C++. Principles of object-oriented programming, as embodied in C++. Introduction to elementary data structures and algorithmic analysis. Core competencies in software engineering, including programming style, optimization, debugging, compilation, program management, development environments, software repositories, version control.

3 References: Books

- Bjarne Stroustrup, The C++ Programming Language (4th Edition), Addison-Wesley, 2013: The author is the creator of C++. This is a definitive reference.
- Y. Daniel Liang, Introduction to Programming with C++, Pearson 3rd edition, 2013. 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.
- 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++.
• Stephen Prata, C++ Primer Plus (6th edition) Developers Library / Addison-Wesley


4 References: Web

• TutorialsPoint C++
• Wikibooks C++ Programming
• cplusplus.com – The C++ resources network
• C++ reference (in English and many other languages)

5 Online Resources

All questions about the course should be asked at the piazza discussion forum: EC327 Piazza

This is also where announcements will be made. You can choose to be notified by piazza via email if anything changes on the site. Click on the gear symbol and go to “Account/Email” settings.

The course materials and submission system are available here: EC327 Course Website

6 Grading and Assessment

6.1 Collaboration policy

The University has very clear policies regarding academic honesty. It considers plagiarism and other forms of cheating serious offenses and will enforce serious penalties when they occur. All students are required to abide by all applicable policies and regulations on academic honesty.

Here is the academic conduct code for Boston University

6.2 Grading Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>15%</td>
</tr>
<tr>
<td>Assignments</td>
<td>50%</td>
</tr>
<tr>
<td>Lab &amp; Participation</td>
<td>10%</td>
</tr>
<tr>
<td>Timed Tests</td>
<td>25%</td>
</tr>
</tbody>
</table>
7 Labs

Each week, there will be four lab sessions and/or office hours as follows:

- Hammas: lab/office hours Tu 12:20-2:05pm
- Hammas: lab/office hours Tu 3:30-5:15pm
- Pouya: lab/office hours W 4:30-6:15 pm
- Pouya: lab/office hours F 2:30-4:15pm

We will also add a session in the evening (TBD).
The zoom links for these will be posted on piazza.
Attendance during the lab sessions is encouraged but not mandatory.

8 Expectations of Achievement Levels

8.1 A-level student

This student consistently delivers A-level software, which means the software

- is well designed
- is well documented
- is conformant to style and formatting guidelines
- achieves all customer requirements
- is delivered on time

The student takes an active and positive role in the class community, often also involving helping other students, discovering specifications that need clarification.

Please note that achieving an A is very difficult and is reserved for students with truly exceptional performance in all aspects of the course. Typically, three times as many A- grades are awarded as full A grades.

8.2 B-level student

This student sometimes achieves the standards for A-level software, and when that level is not achieved, most other software delivered is B-level, which means the software

- does not contain serious design flaws
- is mostly conformant to style and formatting guidelines
- achieves primary customer requirements (i.e. basic functionality, but fails checks)

The student

- takes an active and positive role in the class community, and
- demonstrates effort and commitment to improving their software design techniques
8.3 C-level student

This student rarely achieves A-level software (usually only on the simplest of tasks), and sometimes achieves B-level software.

The rest of the software delivered is C-level software, which means the software is in one of three stages

- compiles and meets some customer requirements
- compiles but meets no other customer requirements, instead having pseudocode/shell programs
- does not compile, has logic or design flaws, but is otherwise “complete”

This student

- demonstrates effort and commitment to improving
- demonstrates basic understanding of fundamentals of software development: syntax, algorithms, program organization
- is able to design algorithms/pseudocode for incomplete or incorrect software
- takes a reasonably active role in the class community (i.e. comes to class most of the time, submits questions on discussion board)

8.4 D or F level student

This student rarely achieves A-level or B-level software, and often submits nothing at all or just bare sketches of programs.

This student

- never demonstrates basic knowledge of syntax, program organization, or good design principles
- does not demonstrate commitment to learning
- does not participate to a satisfactory level in class activities