Course Information and Policy

Administrative
Staff
Office Hours
Course Content
Prerequisite

Resources
References
Textbooks
Optional reference texts
External

Course
Java
Labs
Online

Grades
Composition
Lecture Notes
Homeworks
Group Project
Quizzes
Final Exam
Extra credit

Collaboration

Administrative

Staff

- *Instructor*: Ari Trachtenberg (trachten@bu.edu, 358-1581, PHO 427).
- *Graduate Teaching Assistant*: Hao Yu (imhaoyu@bu.edu, PHO 307).
- *Lab Assistant*: 
John Burke (jwburke@bu.edu, PHO 307),

- **Graders:**
  - Lisa Korver (lkorver@bu.edu)
  - Jia Wilkins (jcw0525@bu.edu)

**Office Hours**

<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
<th>Staff</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sundays</td>
<td>7-8pm</td>
<td>JohnBurke</td>
<td>PHO 307</td>
</tr>
<tr>
<td>Mondays</td>
<td>10-11:30am</td>
<td>HaoYu</td>
<td>PHO 307</td>
</tr>
<tr>
<td>Tuesdays</td>
<td>10:30-11:30am</td>
<td>AriTrachtenberg</td>
<td>PHO 427</td>
</tr>
<tr>
<td>Wednesdays</td>
<td>6:30-8:30pm</td>
<td>JohnBurke</td>
<td>PHO 307</td>
</tr>
<tr>
<td>Thursdays</td>
<td>10-11:30am</td>
<td>HaoYu</td>
<td>PHO 307</td>
</tr>
<tr>
<td>Thursdays</td>
<td>3-4pm</td>
<td>AriTrachtenberg</td>
<td>PHO 427</td>
</tr>
</tbody>
</table>

**Course Content**

Data structures are widely employed to store data in an efficient manner, and can improve computational efficiency by orders of magnitude. The course will be structured into four submodules: review, analysis, data structures, and applications:

**Review**

We will review the fundamentals common to the course material: modern Java language usage, basic analysis techniques for algorithms and data structures, core data structures, and theory (graphs, matrices, combinatorics, and probability).

**Analysis**

We will examine and analyze different approaches of algorithmic design: randomization, sub-linearity, amortization, greed, and dynamic programming.

**Data structures**

We will consider more advanced data structures, such as those used for hashes, filters and sketches, balanced trees, priority queues, and disjoint sets.

**Applications**

We will then consider important relevant applications, such as string matching, machine learning, computability, and approximation.

Throughout, the course attempts to evenly balance rigorous analysis with practical implementation and application to the state of the art using modern software engineering frameworks.

**Prerequisite**

The formal prerequisite for this course is **EC327 - Introduction to Software Engineering** or an equivalent.
serious undergraduate course that utilizes a modern low-level object-oriented language (e.g. C++, C#, Rust, Java). It is further expected that students are familiar with the core undergraduate engineering math curriculum, including calculus and probability, and it is advantageous, but not formally required, to have taken an undergraduate algorithms course.

Resources

References

Textbooks

Required

- [CLRS] Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms* (Fourth Edition), MIT press, 2022: This is the most complete reference for data structures and algorithms currently in use, and it is found on the bookshelves of many professional engineers. (required)

Recommended


Optional reference texts

- Kleinberg and Tardos, *Algorithm Design* (First edition), Perason, 2005: An alternate text used for this course in other semesters.
- Mark Allen Weiss, *Data Structures & Algorithm Analysis in Java* (Third Edition), Pearson 2011: This is a version of an alternate text used for the undergraduate version of this class some semesters. It is not nearly as comprehensive as our textbook [CLRS], but it is somewhat easier to understand and provides Java code, not just pseudocode.

External

- BU Research Computing has various training modules, such as one on linux.
- The Java tutorial: https://download.oracle.com/javase/tutorial/ - “A practical guide for programmers”
- The Java language specification: https://docs.oracle.com/javase/specs/ - The formal specification for Java.
- Platforms
  - Documentation for the IntelliJ IDE.
  - Documentation for the NetBeans IDE.
  - Tutorial for the Eclipse IDE.
This course is taught through the Java programming language. Unless otherwise stated, you may use any 100% pure Java 17 compiler that you want to implement your programs, although:

- Some homework assignments will be provided as IntelliJ, Netbeans or Eclipse projects (which you would have to translate to your choice environment);
- Work on the Android smartphone may be based on its specific development environment through Android Studio;
- We will not provide class-based support for your personal laptops; you are welcome to use the lab machines, which are maintained by the department, for all your work in the class.

Since Java is (in principle) fully portable, I should be able to compile your source code or read your byte code on any pure Java Virtual Machine (or in ART for Android programs). In the event that you need to write non-Java code, please seek approval from the instructor first.

As part of this course, you also have access to the Signet/VLSI Labs (PHO 305/307), which contain Linux workstations. All workstations should have a variety of IDEs installed (IntelliJ, Netbeans, Eclipse …), which you may use in assignments or your project. If you do not have door access to either PHO 305 or PHO 307, please submit your request through Zaius.

You may also use your own devices to complete your home work, but

- You will be responsible for all installation and support on these devices, and
- Your exams will be completed on the lab computers, so you should at least familiarize yourself with their use.

The labs are available for shared lab hours every weekday from 6:30-8:15pm except Mondays (7:30-8:30pm).

We will utilize a variety of professional web-based technologies in this course, including:

**Course wiki**

https://agile.bu.edu/ec504. You are responsible for checking the course wiki regularly. It will contain handouts, parts of your homeworks, projects, exams, and related materials. The wiki will also eventually allow you to check your grades online.

**Gitlab**

https://agile.bu.edu/gitlab. This will serve as our class git server for version control. We will also use it for issue tracking and peer-review.

**Piazza**
https://piazza.com/bu/spring2024/ec504/. You may post informal questions about course material here.

Grades

All grades will be curved according to the class median. Thus, it is your relative score (compared to the rest of the class) that really matters, rather than your objective score. For a course at this level, I expect to center the median at a B/B+, but the final grade will depend on my assessment of the class as a whole.

Composition

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

- Lecture notes (5%)
- Group Project (10%)
- Homeworks (25%) – ~4-5 total
- Quizzes (30%) - ~4 total
- Second exam (30%) - Tuesday, May 7, 3-5pm in PHO 307.

Lecture Notes

Each student will serve as a scribe for ONE graded lecture note over the course of the semester (see the CourseSchedule topic).

Homeworks

Homeworks are due before the beginning of class.

Group Project

You will complete one semester-long group-based project.

Quizzes

There will be roughly four quizzes, each focusing on one submodule covered in class. Quizzes will be completed individually, and with access only to the course wiki and an online Java reference.

Final Exam

There will be one final exam in the class.

Extra credit

Extra credit opportunities will be given out throughout the semester, typically in class and posted to our wiki.
site. The first complete and clearly correct solution will be accepted - even minor errors and technicalities may preclude assignment of credit. Extra credit points will be added to your raw score, thus potentially having a significant effect on your overall grade.

Collaboration

I take cheating and plagiarism very seriously. You may use ONLY any textbooks or automated web sources (i.e. no human feedback) when completing your homework subject to the following strictly enforced conditions:

1. You must clearly acknowledge all your sources and how they were used near the location where they were applied.
2. You must write all answers in your own words, although automatically produced code is acceptable, when acknowledged.
3. You must be able to fully explain your answers upon demand (and I will demand it!).

You may not use any human resource outside of class (including web-based help services, outside tutors, etc.) in completing your homeworks or project. Obviously, you may not collaborate with anyone on the exam.

Failure to meet any of the above conditions will constitute a violation of academic integrity in this class. If you are not sure whether something is permitted by the course policy, ASK THE INSTRUCTOR! - it is much more awkward to explain your actions after the fact to the college disciplinary committee.

- This page was cached on 06 Feb 2024 - 14:38. Get a fresh version here.