CS 102 Computer Science 2
Spring 2014

Instructor Muller

Syllabus

This course is the second part of the two-part CS101-102 introductory sequence. This course builds on what was learned in CS101 and serves as the gateway to the upper levels of the computer science curriculum. This course covers data structures and algorithms that are at the heart of computing. Topics include stacks, queues, trees and graphs as well as searching and sorting algorithms. We will consider trade-offs between linked and sequential implementation as well as mutable versus immutable data structures.

As in CS101, the only way to truly master this material is to read and write a fair amount of software.

Staff

Instructor: Robert Muller, robert.muller2@gmail.com, office: Maloney Hall 570, hours: Mondays 11AM - 1PM, Tuesdays 1PM - 2:30PM, and by appt. 617-552-3964.

Teaching Assistant: Patrick Lenehan, lenehanp at bc dot edu, office: Fulton 160, hours: Mondays 4-5, Wednesdays 12-1 and Thursdays 3-4.

Teaching Assistant: Timothy Oh, ohti at bc dot edu, office: Fulton 160, hours: Wednesdays 6-8PM and Fridays 10AM-12PM.

Teaching Assistant: Nick Parisi, parisin at bc dot edu, office: Fulton 160, hours: Tuesdays 7-9PM and Thursdays 4-5PM.

Teaching Assistant: Tyler Thorn, thornty at bc dot edu, office: Fulton 160, hours: Wednesdays 11-Noon, Thursdays 7-8 PM, and Saturdays 1-3PM.

Lecture

Lectures meet for 150 minutes each week. Unusually, the instructor will be teaching CS102 on both a 50-minute MWF (at 10) schedule and a 75-minute TR (at 9 and 3) schedule. This will no doubt pose logistical challenges during the semester so your patience is appreciated.

Note: There are a lot of really interesting, even some amazing ideas covered in this class and it is important that the classroom be focussed. For this reason, unless specified otherwise, **laptops should be closed once class starts**. If you have a special need for your laptop during lecture, please see the instructor.

Problem Sets

Each week you will be assigned a problem set. Unless otherwise specified, all problem sets are due on 12PM on the specified due date. The single best indicator of success for computer science is *starting problem sets early.*

Problem sets should be submitted for grading by uploading an appropriately named zip file through the course Blackboard/Vista web site. (As linked from the course home page.) Problem sets cannot be submitted as email attachments. Attempts to submit problem sets as email attachments will not receive an email reply indicating that the attempted submission failed.

Exams

There will be a midterm exam and a final exam. The midterm exam will be in-class and is closed notes and closed book. You will have 50 minutes to complete the midterm, and 2.5 hours to complete the final. If you require extra time for documented reasons, please let us know.

Note: Although CS102 does not have an official common final exam schedule, we will be scheduling a common final exam for all three sections. The date, time and location will be announced as soon as we are allocated space by the office of the registrar.

Reading

Algorithms (Fourth Edition, Fourth Printing) by Robert Sedgewick and Kevin Wayne, is the primary text. Also, in reference to Java programming skills we will rely heavily on **Effective Java (Second Edition)**, by Joshua Bloch. Both are available via Amazon.

Grading

Your grade for this class will be a combination of your homework, exam, and class participation. Participation is largely based on effort (not correctness). Final grades are computed, roughly as follows:

- Problem sets, these account for 50% of your grade,
- Midterm exam, accounts for 20% of your grade and a final exam which accounts for 20% of your grade,
- Class participation accounts for the remaining 10% of your grade. By "participation" we mean contributing to the common cause of the course by posing or answering questions in class and by participating in the Piazza forum.

The instructor reserves the right to adjust the relative percentages of the grade components for an individual student if there is a significant discrepancy between the exam scores and the problem set scores.

Late Homework Policy

Homework is due on the day indicated at 12PM, i.e., **noon and not midnight**. *This is a strict deadline.* Homework submitted at 12:01PM is one day late as is homework submitted 23:59 late. Late homework is penalized 25% per 24-hour period.

In the case of medical exigencies, students may petition for an extension. Medical problems or family emergencies are the only conditions under which extensions will be granted.

Honor Code

All solutions and code should be produced by you alone, or by you and a partner, where appropriate. For pair-programmed assignments, each partner needs to submit the assignment and each needs to acknowledge the other partner when submitting.

You may discuss algorithms at a high level with any student in the class. You may also help any student find a small bug in their code. However, you may not copy solutions from anyone, nor should you collaborate beyond high-level discussions with anyone who is not your partner. For pair programming problems, you must follow the guidelines given above.

If you have any questions about what behavior is acceptable, it is your responsibility to come see one of the instructors before you engage in this behavior. We are more than happy to answer any questions you may have.

Getting Assistance

CS102 is a fairly challenging course with programming assignments to match so you may find yourself in need of assistance. The best way to get help on something is to visit a staff member during their office hours. If you cannot make it to office hours you may be able to get your question answered by posing it to the class Piazza forum, you can do so anonymously if you wish.

Generally speaking, sending Java code to a staff member with questions such as "This should work. Why doesn't it?", is not a good way to receive assistance for the required work.

Topics

This the the rough outline. Adjustments and course corrections will be made on a continuing basis.

1. Overview, Course Admin, introduction to Java, Dr. Java, introduction to types, ADTs and APIs, Java interfaces and classes, storage and storage diagrams.
2. Variations on the Stack ADT, Introduction to Java, The Stack ADT, sequential and linked representation, polymorphism, iterators and collections, Strings, StringBuilder and toString, Example application: Dijkstra's two-stack algorithm for expression evaluation.
3. The Queue ADT, sequential and linked representations,
4. (a) Deques,
   (b) List ADTs : `isEmpty`, `length`, `append`, `add`, `rest` and `find`
   (c) Mutable .vs. immutable types,
   (d) Algorithmic complexity.
5. (a) Comparing, equality, == and equal, sets, relations, orders, Java's Comparable and Comparator interfaces, consistency and a warning about Java.
   (b) Sorting algorithms : mergesort, quicksort and insertion sort.
6. More sorting algorithms.
7. (a) Trees, terminology, properties and tree traversals,
   (b) **Midterm Exam**
8. **Spring Break!**
9. Priority Queues, binary heaps, Heapsort.
10. Maps/Symbol Tables/Dictionaries
11. Binary Search Trees.
12. Balanced Search Trees.
13. Game Trees, minimaxing.
14. Hash Tables.
15. String Matching, Regular Expressions.