CAS CS 112: Introduction to Computer Science II
Boston University, Spring 2025

Syllabus

Description: The second course for computer science majors and anyone seeking a rigorous introduction. Covers advanced programming techniques and data structures using the Java language. Topics include searching and sorting, recursion, algorithm analysis, linked lists, stacks, queues, trees, and hash tables. Carries MCS divisional credit in CAS. Fulfills a single unit in each of the following BU Hub areas: Quantitative Reasoning II, Creativity/Innovation, Critical Thinking.

Prerequisites: CAS CS 111, or the equivalent. If you have not had significant prior experience with recursion, you are strongly encouraged to take CS 111 first.

Instructor(s):
Christine Papadakis-Kanaris (cpk@bu.edu, CDS 943)

Undergraduate Teaching Assistant(s) (TA)
Oliver Bailey (livfernb@bu.edu)
Rachel Cherry (rcherry@bu.edu)
Georgina Foccia (gfocia@bu.edu)
John Muenning (jmue@bu.edu)
Sophia Howson (showson@bu.edu)
Toby Ueno (tobyueno@bu.edu)

Course Assistants (CAs)
We are fortunate to have several undergraduate course assistants (CAs) as members of the course staff. They will be working with you in the labs and holding office hours each week. See the course website for their names and contact info.

Lectures and Labs
lectures: section A1: T/Th 9:30 a.m. – 10:45 a.m. CGS 129
             section B1: TuTh, 11:00 am-12:15 pm, CGS 129

lab: a weekly, one-hour session; see your schedule for the time and location
Important: You must also be able to take the midterm exams, which will be held on two Wednesday evenings from 6:30-8:00 p.m. We are not scheduled to meet on any other Wednesday evenings, but may use this time for supplemental tutorials.

Course Website: http://www.cs.bu.edu/courses/cs112
In addition, announcements and some course materials will be posted Blackboard.

Requirements and Grading
1. Weekly problem sets (20% of the final grade)
2. Exams: two midterm exams (35%) and a (*cumulative*) final exam (40%)
3. Quizzes and Participation (5%)

To pass the course, you must earn a passing average across the three exams.

Collaboration Policy
You are strongly encouraged to collaborate with one another in studying the lecture materials and preparing for quizzes and exams. Problem sets will include:
- *individual-only* problems that you must complete on your own, and
- *pair-optional* problems that you may complete alone or with a partner.

For both types of problems, you may discuss ideas and approaches with others (provided that you acknowledge this in your solution), but such discussions should be kept at a high level and should not involve actual details of the code or of other types of answers. **You must complete the actual solutions on your own** (or, in the case of a pair-optional problem, with your partner if you choose to use one).

Rules for working with a partner on pair-optional problems:
- You may *not* work with more than one partner on a given assignment. (However, you are welcome to switch partners between assignments.)
- **You may *not* split up the work and complete it separately.**
- **You must work together** (at the same computer or via a Zoom meeting) for all problems completed as a pair, and your work must be a collaborative effort.
- You and your partner must *both* submit the same solution to each problem that you did as a pair, and you must clearly indicate that you worked on the problem as a pair by putting your partner's name at the top of the file.

Academic Misconduct
We will assume that you understand BU's Academic Conduct Code and have carefully review the CS department's page on academic integrity:
http://www.bu.edu/academics/policies/academic-conduct-code
http://www.bu.edu/cs/undergraduate/undergraduate-life/academic-integrity

Prohibited behaviors include:
- copying all or part of someone else's work, even if you subsequently modify it; this includes cases in which someone tells you what to write for your solution
- with the exception of your partner on pair-optional assignments, you cannot:
  - view all or part of someone else's work
  - show all or part of your work to another student
- consulting solutions from past semesters, or those found online or in books
- using tools that automate (e.g. ChatGPT) and/or assist with the writing of code or the completion of other types of solutions
- posting your work online or where others can view it - even after you complete the course
- **receiving assistance from others or collaborating with others during an exam, or consulting materials except those that are explicitly allowed. Using ChatGPT, GitHub Copilot or any other on-line source for solutions.**

Incidents of academic misconduct may be reported to the Academic Conduct Committee (ACC), and the ACC may suspend/expel students found guilty of misconduct. **At a minimum, students who engage in misconduct will receive a score of 0 on the assignment or exam in question.**

Other Policies

Writing code: The problem sets will require you to write code using the Java programming language. In doing so, you must limit yourself to aspects of the language that we have discussed in lecture, unless the problem indicates otherwise. At a minimum, failure to do so will result in a score of 0 for the corresponding problem.

Laptops: Students taking CS courses are expected to have a laptop capable of running a currently supported version of Microsoft Windows, Mac OS X, or Linux. See this page for more info: https://www.bu.edu/cs/undergraduate/undergraduate-life/laptops

Late problem sets: Problem sets will be divided into two parts, *Part I* and *Part II*. Both parts of the problem set will have the same stated due date, but separate submission portals through Gradescope. **We will not accept any submissions for Part I beyond the stated deadline.** However, the submission portal for Part II submissions will remain open for a 24-hour grace period after the posted deadline. There will be NO deduction for submissions within the grace period, however, students who *consistently* submit their Part-II submission by the stated deadline will receive extra credit towards their problem set score (instructor discretion). Plan your time carefully, and don't wait until the last minute so you will have ample time to ask questions and obtain assistance from the course staff.

Pre-lecture preparation: To help you prepare for lecture, you are encouraged to complete an assigned reading. This preparation is not graded, but failing to complete it will make it more difficult for you to understand the material presented in lecture.

Post-lecture review: To help you assess your understanding of the material presented in lecture, there will be a weekly on-line post-lecture assignment that will be posted at the end-of-day Thursday and will be due by the end-of-day the following Monday. There will not be any late submissions for these assessments and these assessments will be graded on participation **and** correctness.

The *attendance/participation* portion of your grade will be based on your consistent attendance at the lectures and lab sessions. Attendance will be taken at every lab, and you must attend at least 85% of the lab sessions to get full credit for lab participation. We will also periodically take attendance at lecture, and you must be present for 85% of the lectures at which attendance is taken to get full credit for lecture participation.

The final grades are *not* curved. The performance of the class as a whole is taken into account in assigning letter grades, but this can only improve your grade, not harm it.

Extensions and makeup exams will only be given in documented cases of serious illness or other emergencies.

You cannot redo or complete extra work to improve your grade.

Incompletes will not be given except in extraordinary circumstances.

Course Materials

Textbook: You are not required to purchase a textbook. Instead, we will be assigning readings from freely available online resources. If you are interested in purchasing a Java reference book, we will recommend some possible titles in lecture.

In-class software: We will be using the Top Hat platform for in-class activities and attendance. More detail will be provided in class.

Schedule (tentative)

Week	lecture dates	topics, exams, assignments, and special dates
0	1/21, 1/23	Course overview and introduction Java basics; Conditional execution and user input
1	1/28, 2/30	More on conditional logic, static methods; loops; variable scope
2	2/4, 2/6	Primitives, objects, and references; Java Arrays
3	2/11, 2/13	Object-oriented programming (classes and objects in Java); Memory management
4	2/20	Inheritance and polymorphism;
5	2/25, 2/27	Wrapping up Inheritance; A simple collection class Recursion; *2/25: last day to drop without a 'W'*
6	3/4, 3/5*, 3/6	Recursive Backtracking; Algorithm Analysis & Big-O notation. ****Midterm 1 on 3/5 from 6:30-8:00 pm***
	3/8-3/16	*Spring break*
7	3/18, 3/20	Wrapping up with Sorting; Linked lists
8	3/25, 3/27	Abstract data types (ADTs) and interfaces List ADT; iterators *3/29: last day to drop a class with a 'W'*

9	4/1, 4/3	Stack and queue ADTs and Java generics *4/4: last day to drop with a 'W' or change to Pass/Fail*
10	4/8, 4/9*, 4/10	Tree basics; binary trees ***Midterm 2 on 4/9 from 6:30-8:00 pm**
11	4/15, 4/17	Search Trees; Balanced search trees Hash tables
12	4/22, 4/24	Hash tables (cont.) Heaps and priority queues
13	4/29, 5/1	Heaps and priority queues (cont.) Heapsort Semester Wrap-up *5/2-5/4: Study period*
14		**Final exam: time and date TBD** **Please wait until your instructor informs you of the date. The initial date posted by the Registrar may <u>not</u> be correct.** ***Make sure that you are available for the entire exam period – up to and including Friday evening, May 9th***

Please note above Schedule is (tentative and subject to change)