CAS CS 112: Introduction to Computer Science II  
Boston University, Summer I 2020

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.

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  
Christine Papadakis-Kanaris (cpk@cs.bu.edu)

Teaching Fellow  
Anam Farrukh (afarrukh@bu.edu)  
Cheng Zhang (czhang03@bu.edu)  
Athanasios Filippidis (aflpd@bu.edu)

Tutor  
Tristan Marshand (tmarch@bu.edu)

Meeting Times and Places  
lecture: MTWTh, 10:00 a.m. - 12:00 p.m., virtual  
lecture: MTWTh, 1:00 – 3:00 p.m., virtual  
lab: W 8:00-10:00 a.m., virtual  
lab: W 10:00 a.m. – 12:00 p.m., virtual

Course Website: [http://www.cs.bu.edu/courses/cs112](http://www.cs.bu.edu/courses/cs112)

Requirements and Grading  
1. Exams: midterm exam (30%) and a final exam (35%)  
2. Problem sets (25%)  
3. Quizzes (5%)  
4. Class Participation (5%)

To pass the course, you must earn a passing grade on the final exam and the problem sets.

The percentages shown are guidelines and may be changed to better reflect course content and delivery.
Collaboration Policy
You are encouraged to collaborate with one another in studying the lecture materials and preparing for quizzes and exams. During this summer session all collaboration that is done with respect to this course must be virtual. There will be no person to person contact as related to this course, even for those students who are on or near campus.

Problem sets may include two types of problems:
- *individual-only* problems that you must complete on your own
- *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 for every problem that you complete as a pair.** While you are working, the screen should be visible to both of you. One person should type, while the other person observes, critiques, and plans what to do next. You should switch roles periodically, and your solution should be a true collaborative effort.
- You 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:
[http://www.bu.edu/academics/policies/academic-conduct-code](http://www.bu.edu/academics/policies/academic-conduct-code)

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 you should write for your solution
- viewing all or part of someone else's work (with the exception of work that you and your partner do together on a pair-optional problem)
- showing all or part of your work to another student (with the exception of work that you and your partner do together on a pair-optional problem)
- consulting solutions from past semesters, or those found online or in books
- posting your work where others can view it (e.g., online).

Incidents of academic misconduct will be reported to the Academic Conduct Committee (ACC). The ACC may suspend/expel students found guilty of misconduct. **At a minimum, students who engage in misconduct will have their final grade reduced by one letter grade (e.g., from a B to a C).**
Other Policies

*Late problem sets:* Problem sets must be submitted by the date and time listed on the assignment.

The final grades are *not* curved. The performance of the class as a whole is taken into account when 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.**

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)

<table>
<thead>
<tr>
<th>Week</th>
<th>Lecture dates</th>
<th>Topics, exams, assignments, and special dates</th>
</tr>
</thead>
</table>
| 1    | 5/20, 5/21    | Course overview and introduction; Java basics: control structures, static methods, and data types  
|      | NO LAB meeting | Problem Set 1 due 5/28 |
| 2    | 5/25* no class, 5/26, 5/27, 5/28 | Java Loops; Scope of Variables; Primitives, References and the Java memory model Java arrays  
|      | 5/27 LAB @ 8-10 a.m., 10-noon | Problem Set 2 due 6/3 |
| 3    | 6/1, 6/2, 6/3, 6/4 | Object-oriented programming; Classes and Objects; Inheritance and Polymorphism; Array Bag case study; Recursion.  
|      | 6/3 LAB @ 8-10 a.m., 10-noon | Problem Set 3 due 6/8 |
| 4    | 6/8, 6/9, 6/10, 6/11 | Recursive Backtracking; Introduction to Algorithmic Analysis; Sorting and algorithm analysis; Linked Lists, interfaces and iterators;  
|      | 6/10 LAB @ 8-10 a.m., 10-noon | Problem Set 4 due 6/12 |
| 5    | 6/15*, 6/16, 6/17, 6/18 | Stack and Queue ADT; generic interfaces; Binary Trees; Search Trees (an introduction) Search Trees cont’d; Balanced Trees (2-3 trees, B-trees); Complete Trees.  
|      | 6/17 LAB @ 8-10 a.m., 10-noon | Midterm Exam – through material covered 6/11 |
| 6    | 6/22, 6/23, 6/24, 6/25 | The Heap; Hash Tables;  
|      | 6/24 LAB @ 8-10 a.m., 10-noon | Problem Set 5 due TBA |
| 7    | 6/29, 6/30, 7/1* | Priority Queue, HashSort, Huffman Codes, Class wrap-up  
|      | 5/27 LAB @ 8-10 a.m., 10-noon | Problem Set 6 due (morning of) 6/30 |
| 7/1  | Final Exam    |                                           |