Computer Science S-111
Intensive Introduction to Computer Science

Harvard Summer School 2020

Syllabus

Overview
This course is a fast-paced and rigorous introduction to computer science. The first half of the course covers foundational programming concepts such as data types, conditional execution, iteration, and recursion. It also explores the key features of object-oriented programming, as well as the manipulation of data stored in files and arrays. The second half of the course provides a survey of fundamental data structures including lists, stacks, queues, trees, and graphs. It explores the implementation of these data structures using both array-based and linked representations, and it examines classic algorithms that use these structures for tasks such as sorting, searching, and text compression. Techniques for analyzing the efficiency of algorithms are also covered.

Problem sets require a minimum of twenty hours of work each week, including both written problems and programming exercises using the Java programming language. Graduate-credit students are expected to complete additional work. The course includes coverage of the key topics needed for the AP Computer Science A examination. For Harvard College students, the course counts as 8 degree credits but only 4 concentration credits (equivalent to Computer Science 50).

Prerequisites
Familiarity with precalculus. No prior programming experience is required. Students who have completed the Harvard Extension School courses CSCI E-10a, CSCI E-10b, CSCI E-22, or CSCI E-50 (CSCI-50a and CSCI E-50b) cannot earn degree credit for CSCI S-111.

Instructor
David G. Sullivan, Ph.D. (dgs@cs.bu.edu)
Master Lecturer on Computer Science, Boston University
office hours: see the course website

Teaching Assistants (see the course website for office hours)
Ashby Hobart (ahobart@bu.edu)
Libby James (etjames@bu.edu)

Meeting Times and Places

Lectures: Mon-Fri, noon-3 pm Eastern time. This course meets via live web conference. Students must attend and participate at the scheduled meeting time. There is one day each week without lecture – usually Wednesday, but there are exceptions. See the schedule below for more detail.

Sections: daily from 4-5 pm via live web conference on days when lecture is held. Attendance at both the lectures and sections is essential, as this course moves very fast. We also encourage you to meet regularly with a member of the teaching staff to review any problems you are having with the homework or specific topics.

Requirements

The course is divided into ten distinct units. Units 1-5 cover programming fundamentals, and units 6-10 cover data structures and algorithms.

1. *Problem sets:* Each unit has a problem set consisting of two parts. Part I typically consists of short "written" problems that test your understanding of the key concepts from the unit. Part II consists of one or more programming problems that require you to employ the concepts from the unit. All programming problems must be completed in Java, and ***they must compile and run in order to be eligible for full credit***.
2. *Unit tests:* At four points during the course (see the schedule below), students will take a one-hour test on material from the prior two units of the course.
3. *Final exam:* a three-hour comprehensive exam at the end of the course.

Important note: The problem sets tend to be **extremely** time-consuming. Don't wait until the last minute to begin them! You should plan on devoting approximately 20-30 hours of work per week. ***If you have other major time commitments, you should reconsider whether to take this course.***

Graduate-credit students: Students taking the course for graduate credit must complete additional homework. On most problem sets, the problems required of all students will be worth a total of 100 points; grad-credit students will complete one or two additional problems worth a total of 10 points. These grad-credit problems are typically more challenging than the other problems, and thus grad-credit students should plan to spend approximately 20% more time on the homework.

Grading Policies

Late penalties: Homework is due **by 11:59 p.m.** on the date listed on the assignment. There will be a 10% deduction for homework that is up to 24 hours late. ***We will not accept any homework that is more than 24 hours late.*** Plan your time carefully, and don't wait until the last minute to begin an assignment. Starting early will give you ample time to ask questions and obtain assistance.

Determining the final grade: homework 40%, unit tests 25%, final exam 35%

The final exam will replace your lowest assignment grade if doing so helps your final grade. The final exam will also replace your lowest unit-test grade if doing so helps your final grade.

A letter grade will be given in accordance with the Summer School's grading policy (see https://www.extension.harvard.edu/grades). 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 makeups will only be given in *documented* cases of serious illness or other emergencies. You cannot redo or complete extra work to improve your grade.

An EXT (extension) grade will be granted only in extreme circumstances (e.g., illness), and only when appropriate documentation has been provided. Please bring any such circumstances to Dr. Sullivan's attention as soon as possible.

Academic Conduct

Rules for working on the problems:

- You may discuss the main ideas of a given problem with other students (provided that you acknowledge doing so in your solution), but you must complete the actual solution by yourself.
- You may *not* copy all or part of another person's work, even if you subsequently modify it.
- You may *not* view all or part of another student's work.
- You may *not* show all or part of your work to another student.
- You may *not* consult solutions from past semesters, or those found in books or on the Web.

If we believe that a student is guilty of academic misconduct, we will refer the matter to the Administrative Board of the Summer School, who could require withdrawal from the course and suspension from all future work at the School.

The Summer School provides resources to support academic integrity here:
http://www.summer.harvard.edu/resources-policies/resources-support-academic-integrity

Summer School Policies

We also expect you to know and adhere to the general policies and procedures of the Summer School: http://www.summer.harvard.edu/policies/student-responsibilities

Accessibility Services

The Summer School is committed to providing an accessible academic community. The Accessibility Services Office offers a variety of accommodations and services to students with documented accessibility issues. This site has more information:
https://www.summer.harvard.edu/resources-policies/accessibility-services

Textbooks

- CSCI S-111 coursepack. This contains all of the lecture notes for the course. An electronic copy will be available on the course website, and we encourage you to print it so that you can mark up the pages during lecture.
- **Optional:** *Building Java Programs, 4th ed.* by Stuart Reges and Marty Stepp (Pearson, 2016). Older versions are also fine. This book is *not* required.
- **Optional:** *Data Structures & Algorithms in Java, 2nd ed.* by Robert Lafore (SAMS Publishing, 2003). This book is *not* required.

Course Outline

Unit 1: Getting started. Programming in Scratch (a graphical language developed at MIT that will allow us to quickly introduce a number of key programming concepts). Simple Java programs. Statements. Standard output. Procedural decomposition using simple methods.

Unit 2: Imperative programming, part I. The programming process. Data types. Literals, variables, and expressions. Definite loops. Simple conditional execution.

Unit 3: Imperative programming, part II. Methods with parameters and return values. Using objects. Strings. Console input using Scanner objects. Conditional execution revisited. Indefinite loops and boolean expressions.

Unit 4: Processing collections of data. Arrays. References and reference variables. File processing. A first look at recursion.

Unit 5: Object-oriented programming. Writing "blueprint" classes. Fields, non-static methods, and constructors. Inheritance and polymorphism.

Unit 6: Foundations of data structures. Defining and implementing an abstract data type. Memory allocation (stack and heap storage). Recursion revisited, including recursive backtracking algorithms.

Unit 7: Sorting and algorithm analysis. Sorting arrays using the following algorithms: insertion sort, selection sort, bubble sort, Shellsort, quicksort, and radix sort. Algorithm analysis: running-time analysis; big-O notation; worst-case, average-case, and best-case analyses.

Unit 8: Sequences. Linked lists. List, stack, and queue abstract data types, including both array and linked-list implementations of each of these ADTs. Implementing a generic collection.

Unit 9: Trees and hash tables. Tree overview and terminology. Data structures and algorithms for data dictionaries: binary search trees, 2-3 trees, B-trees, and hash tables. Data compression using Huffman encoding. Heaps and priority queues.

Unit 10: Graphs. Graph overview and terminology. Depth-first and breadth-first traversals. Dijkstra's shortest-path algorithm. Minimum spanning trees. Topological sort. Classifying problems.

Schedule (tentative) **key:** BJP = *Building Java Programs;* L = Lafore book

	lecture date	unit(s) covered	due dates	unit tests	optional readings
Mon	June 22	Unit 1			
Tues	June 23	Unit 1			BJP 1
Wed	June 24	*No lecture*	PS 1, part I due		
Thurs	June 25	Unit 2	PS 1, part II due		BJP 2, 4.1
Fri	June 26	Unit 2			
Mon	June 29	Unit 3	PS 2, part I due		BJP 3
Tues	June 30	Unit 3	PS 2, part II due		
Wed	July 1	Unit 3			BJP 4
Thurs	July 2	Unit 3		Units 1&2	BJP 5.1-4
Fri	July 3	*No lecture*			
Mon	July 6	Unit 4	PS 3, part I due		BJP 7
Tues	July 7	Unit 4	PS 3, part II due		BJP 6
Wed	July 8	Unit 4			BJP 12
Thurs	July 9	Unit 5	PS 4, part I due		BJP 8
Fri	July 10	*No lecture*			
Mon	July 13	Unit 5	PS 4, part II due		BJP 9
Tues	July 14	Unit 5			
Wed	July 15	*No lecture*	PS 5, part I due		
Thurs	July 16	Unit 6		Units 3&4	L 1
Fri	July 17	Unit 6	PS 5, part II due		L 6
Mon	July 20	Unit 7	PS 6, part I due		L 2, 3, 7 (B 13)
Tues	July 21	Unit 7			
Wed	July 22	*No lecture*	PS 6, part II due		
Thurs	July 23	Unit 8			L 4, 5 (B 16, 15)
Fri	July 24	Unit 8	PS 7, part I due	Units 5&6	(B 14)
Mon	July 27	finish unit 8 start Unit 9	PS 7, part II due		L 8 (B 17)
Tues	July 28	Unit 9	PS 8, part I due		L 10
Wed	July 29	*No lecture*			
Thurs	July 30	Unit 9	PS 8, part II due		L 12 (B 18)
Fri	July 31	finish Unit 9 start Unit 10	PS 9, part I due		L 11
Mon	August 3	Unit 10	PS 9, part II due	Units 7&8	L 13
Tues	August 4	Unit 10			
Wed	August 5	Wrap-up/ review	PS 10 due		
Thurs	August 6	*No lecture*			
Fri	August 7	***Final exam,*** *8:30-11:30 a.m.*			

Other important dates:
June 18: course registration deadline
June 24: last day to drop courses for 100% tuition refund
July 1: last day to drop courses for 50% tuition refund
July 24: last day to withdraw with a grade of WD (no refund)