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 cannot earn degree credit for CSCI S-111.

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

Teaching Assistants
Kristen Lamb (krlamb@bu.edu)
Edwyn Song (esong501@bu.edu)

Course Website
https://cscis111.sites.fas.harvard.edu
Meeting Times and Places

Lectures: Mon-Fri, 8:30-11:30 a.m., Maxwell-Dworkin G115
There is one day each week without lecture – usually Wednesday, but there are
exceptions. See the schedule below for more detail.
Sections: daily one-hour meetings in the early afternoon on days when lecture is
held; times and locations TBA.
Attendance is required. We also encourage you to meet regularly with a member
of the teaching staff to review any problems that you are having with the homework
or with 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 10 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 (https://summer.harvard.edu/policies/academic-policies/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.

**Collaboration Policy**
You are strongly encouraged to collaborate with one another in studying the lecture materials and preparing for the exams. Problem sets will include:

- **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 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**
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
- 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)
- receiving assistance from others or collaborating with others during an exam, or consulting materials except those that are explicitly allowed.
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.

You are responsible for understanding Summer School policies on academic integrity (https://www.summer.harvard.edu/policies/student-responsibilities). Not knowing the rules, misunderstanding the rules, running out of time, submitting the wrong version of an assignment, or being overwhelmed with multiple demands are not acceptable excuses. Resources to support academic integrity can be found here: http://www.summer.harvard.edu/resources-policies/resources-support-academic-integrity

Publishing or Distributing Course Materials
Students may not post, publish, sell, or otherwise publicly distribute course materials without the written permission of the course instructor. Such materials include, but are not limited to, the following: lecture notes, lecture slides, video, or audio recordings, assignments, problem sets, examinations, other students’ work, and answer keys. Students who sell, post, publish, or distribute course materials without written permission, whether for the purposes of soliciting answers or otherwise, may be subject to disciplinary action, up to and including requirement to withdraw from the Summer School. Further, students may not make video or audio recordings of class sessions for their own use without written permission of the instructor.

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. It will be available from Gnomon Copy (1308 Mass Ave., across from the Yard), or you can access electronic copies of the notes on the course website.
- **Optional:** *Building Java Programs, 5th ed.* by Stuart Reges and Marty Stepp (Pearson, 2019). Older versions are also fine. This book is *not* required.
Course Outline


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


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.


### Schedule (tentative)

**key:** B = Building Java Programs; L = Lafore book

<table>
<thead>
<tr>
<th>date</th>
<th>unit(s)</th>
<th>due dates</th>
<th>unit tests</th>
<th>optional rdng.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon June 20</td>
<td>No lecture</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tues June 21</td>
<td>Unit 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wed June 22</td>
<td>Unit 1; Unit 2</td>
<td></td>
<td>B1</td>
<td></td>
</tr>
<tr>
<td>Thurs June 23</td>
<td>Unit 2</td>
<td>PS 1: Part I due</td>
<td>B2, 4.1</td>
<td></td>
</tr>
<tr>
<td>Fri June 24</td>
<td>Unit 2</td>
<td>PS 1: Part II due</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon June 27</td>
<td>Unit 3</td>
<td>PS 2: Part I due</td>
<td>B3</td>
<td></td>
</tr>
<tr>
<td>Tues June 28</td>
<td>Unit 3</td>
<td>PS 2: Part II due</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wed June 29</td>
<td>No lecture</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thurs June 30</td>
<td>Unit 3</td>
<td></td>
<td>B4</td>
<td></td>
</tr>
<tr>
<td>Fri July 1</td>
<td>Unit 3</td>
<td>Units 1&amp;2</td>
<td>B5.1-4</td>
<td></td>
</tr>
<tr>
<td>Mon July 4</td>
<td>No lecture</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tues July 5</td>
<td>Unit 4</td>
<td>PS 3: Part I due</td>
<td>B7</td>
<td></td>
</tr>
<tr>
<td>Wed July 6</td>
<td>Unit 4</td>
<td>PS 3: Part II due</td>
<td>B6</td>
<td></td>
</tr>
<tr>
<td>Thurs July 7</td>
<td>Unit 4</td>
<td></td>
<td>B12</td>
<td></td>
</tr>
<tr>
<td>Fri July 8</td>
<td>Unit 5</td>
<td>PS 4: Part I due</td>
<td>B8</td>
<td></td>
</tr>
<tr>
<td>Mon July 11</td>
<td>Unit 5</td>
<td>PS 4: Part II due</td>
<td>B9</td>
<td></td>
</tr>
<tr>
<td>Tues July 12</td>
<td>Unit 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wed July 13</td>
<td>Unit 6</td>
<td>PS 5: Part I due</td>
<td>L1</td>
<td></td>
</tr>
<tr>
<td>Thurs July 14</td>
<td>Unit 6</td>
<td>Units 3&amp;4</td>
<td>L6</td>
<td></td>
</tr>
<tr>
<td>Fri July 15</td>
<td>No lecture</td>
<td>PS 5: Part II due</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon July 18</td>
<td>Unit 7</td>
<td>PS 6: Part I due</td>
<td>L2,3,7</td>
<td></td>
</tr>
<tr>
<td>Tues July 19</td>
<td>Unit 7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wed July 20</td>
<td>No lecture</td>
<td>PS 6: Part II due</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thurs July 21</td>
<td>Unit 8</td>
<td></td>
<td>L4,5</td>
<td></td>
</tr>
<tr>
<td>Fri July 22</td>
<td>Unit 8</td>
<td>PS 7: Part I due</td>
<td>Units 5&amp;6</td>
<td>B14</td>
</tr>
<tr>
<td>Mon July 25</td>
<td>Units 8 &amp; 9</td>
<td>PS 7, part II due</td>
<td></td>
<td>L8</td>
</tr>
<tr>
<td>Tues July 26</td>
<td>Unit 9</td>
<td>PS 8, part I due</td>
<td>L10</td>
<td></td>
</tr>
<tr>
<td>Wed July 27</td>
<td>No lecture</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thurs July 28</td>
<td>Unit 9</td>
<td>PS 8, part II due</td>
<td></td>
<td>L12</td>
</tr>
<tr>
<td>Fri July 29</td>
<td>Units 9 &amp; 10</td>
<td>PS 9, part I due</td>
<td>Units 7&amp;8</td>
<td>L11</td>
</tr>
<tr>
<td>Mon August 1</td>
<td>Unit 10</td>
<td>PS 9, part II due</td>
<td></td>
<td>L13</td>
</tr>
<tr>
<td>Tues August 2</td>
<td>Unit 10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wed August 3</td>
<td>Review</td>
<td>PS 10 due</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thurs August 4</td>
<td>No lecture</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fri August 5</td>
<td>Final exam</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Other important dates:
- June 16: course registration deadline
- June 23: last day to drop courses for 100% tuition refund
- June 30: last day to drop courses for 50% tuition refund
- July 22: last day to withdraw with a grade of WD (no refund)

David G. Sullivan, Ph.D.  page 6 of 6