

Introduction to Computer Science II with Intensive C++ CAS CS113 (Spring, 2000) *Syllabus as of 1/12/00*

Instructor

Stan Sclaroff (Office hours: Tuesday 2–5)
Room MCS-279, phone 353-8928
sclaroff@cs.bu.edu

Teaching Assistant

Joni Alon (Office hours: TBA, Location: MCS-266)
Room MCS-205, phone 353-5227
jalon@cs.bu.edu

Course Description

The course begins with an intensive introduction to C++ and then covers all the materials in CS 112. You will learn advanced programming techniques involving dynamic memory allocation, pointers, linked lists, stacks, recursion, trees, graphs, and some searching and sorting. All of this will be embedded into the highly disciplined structure provided by the object-oriented programming language C++.

Prerequisites

This course is designed for students who already program proficiently in PASCAL, C, FORTRAN, or some other high-level programming language. **Note:** If you do not have such previous exposure to programming, then you are strongly encouraged to transfer to CS 111.

Work Load

Heavy, fast, challenging, *and rewarding*. This is an intensive course that covers the topics of CS 111 and CS 112 in one semester. There are weekly programming assignments.

Lectures

TR 12:30–2pm, in Room CAS 522

Sections

Section	Time	Location
A2	W 10-11	ENA 304
A3	W 12-1	ENA 304
A4	W 4-5	ENA 304
A5	W 7-8	ENA 304

Lab section meetings convene in the CS computer instruction lab (ENA 304). The main undergrad computer science lab and instruction lab are located above the Radio Shack.

Note: Completion of lab programming exercises contributes towards your final grade.

Required Texts

1. C. Horstmann, *Computing Concepts with C++ Essentials*, 2nd Edition, Wiley 1998, ISBN 0-471-13770-7.
2. M. Main and W. Savitch, *Data Structures and Other Objects Using C++*, Addison-Wesley 1997, ISBN 0-8053-7470-1.

Supplemental Text

S. B. Lippman and J. Lajoie, *C++ Primer*, 3rd Edition, Addison-Wesley 1998, ISBN 0-201-82470-1.

Class Home Page

<http://cs-www.bu.edu/faculty/sclaroff/courses/cs113/Home.html>

All class assignments, schedules, and lecture notes can be found on this page.

Getting Started

You are expected to be familiar with UNIX and EMACS. Some help will be available in the section, but if you have not used UNIX or EMACS before, then you should attend the appropriate tutorials provided by B.U. Office of Information Technology (<http://www.bu.edu/cc/tutorials/>).

Grading

Programming assignments 50%

Labs 10%

Quiz 15%

Final 25% (comprehensive)

If there are questions about grades, please see the TF.

Incompletes will not be given.

Tests

Quiz Thursday March 2, in class

Final Tuesday May 16, 2:00–4:00PM

Tests are closed-book.

Requests for make-up exams will be granted only when the request is supported by an official doctor's note. If you miss the exam due to death of a close relative, please provide a photocopy of the obituary.

Programming Assignments

Programming assignments are due **before class** on the assignment due date.

Note: Under no circumstances will late assignments be accepted. This is a fast-paced course with programming assignments due each week. If you miss the assignment deadline, then move on. Get started on the next assignment.

Collaboration

Violations of the CAS Academic Conduct Code will be reported to the Dean and may result in a course grade of F, academic probation, or expulsion from the University. Copies of the CAS academic conduct code are available in CAS 105.

It is reasonable to discuss with others possible general approaches to problems. It is unreasonable to work together on a solution, to copy a solution, or to give away a solution. If your common discussion can be detected by looking at the solutions, then there is too much collaboration.

Do not allow your work to be used by others:

- Set file permissions so that they are not readable by others.
- Do not recycle old printouts. Take them home to throw out.
- Do not e-mail your solution to anyone.

Warning: If someone cheats by using your work, you will also be penalized.

Computing Resources

You will need a CSA account, which you can open at the CS Lab (730 Commonwealth Avenue, above Radio Shack). Note that even if you already have an account on the ACS cluster, you still need to open a new account for CSA, which is a different machine.

You may use any C++ compiler to do your work, however the code that you submit must compile and run on CSA with gnu g++. You may access CSA from home if you have a modem or ethernet connection; if not, you may use the CS Lab or the ACS lab in the basement of MCS.

Course Outline (subject to change)

Week of	Lecture	Readings	Section	Notes
Jan	10	Introduction	H 1	no lab
	17	Fundamentals of C++	H 2,3	unix, compilation
	24	Flow of control, functions	H 4,5	objects
	31	Iteration, recursion, debugging	H 6,7	functions, debugging
Feb	7	Classes	H 8	classes
	14	Vectors, arrays, streams	H 9	arrays, debugger
	21	Running time analysis	MS Appendix B	review
	28	Program specification	MS 2	no lab
<i>spring break</i>				
Mar	13	Containers, pointers, dynamic memory	MS 3,4	O-notation
	20	Templates, linked lists	MS 5,6	linked lists
	27	Stacks, queues	MS 7,8	stacks, queues
Apr	3	Trees, recursion	MS 9,10	trees, recursion
	10	Searching	MS 12	trees and sorting
	17	Sorting	MS 13	sorting
	24	Graphs	MS 15	graphs
May	1	Review		Discussion: review

H = Horstmann text.

MS = Main and Savitch text.