

# CSCI 360, Fall 2004

Survey of programming languages  
www.cburch.com/cs/360/

**Instructor:** Dr Carl Burch  
**E-mail:** burch@grendel.hendrix.edu  
**Telephone:** 450-1377 (office); 548-6135 (home)  
**Office:** M. C. Reynolds 310  
**Hours:** M 10a-11a, T 3p-4p, W 3p-4p, F 10a-11a  
drop-ins, appointments always welcome

## Goal

By the end of this course, you should be familiar with examples from the three major categories of modern programming languages — functional, procedural, and object-oriented. You will also have a rudimentary understanding of the process underlying compilation and interpretation, as well as the theory underlying programming languages.

## Textbooks

I will rely on handouts, primarily written by me, but also taken from the Web.

## Evaluation

There are a total of 1,000 points over the semester.

Class attendance/participation	100 pts
Assignments and projects (variable)	500 pts
Quizzes (three, 50 pts each)	150 pts
Midterm	100 pts
Final	150 pts
<b>Total</b>	<b>1,000 pts</b>

Letter grades will come from the following scale.

A	900 or more	D	600 to 699
B	800 to 899	F	less than 600
C	700 to 799		

I reserve the right to make adjustments in the entire grading scheme or in particular cases.

The A grade is meant for outstanding performance; a B is more typical and represents solid preparation for future computer science courses, while a C represents marginally adequate preparation. I assign grades independent of any overall goal, but the average letter grade will likely be near 3.0. Bottom line: B isn't bad.

Note that I do not normally “curve” in determining course letter grades; instead, I monitor your progress and

perform any “curves” as I grade tests. When I “curve” scores, I add a fixed amount to all scores; as a result, a high test score may end up being above 100%. I anticipate, but will not insist, that the average test score will be around 75%. Normally, scores in the other categories will be higher, so that the average class grade will be around B even though the average test grade is around C.

## Class attendance/participation

Several points are designated for “class participation.” I will assign half of these points near the semester’s middle, and the other half near the semester’s end. I *may* use some points for in-class exercises, but most of the points will be based on attendance and participation.

I will monitor your class attendance. If your attendance is excellent (missing one or fewer classes each half-semester), you will receive at least 60% of these attendance/participation points. If you feel your absence should be excused, please warn me about the absence a day in advance. Whether I excuse your absence is my call.

The remaining 40% of these points are for participation, including both questions during class and responses to questions during class. I may give more than full credit in unusual circumstances. Take this as an invitation: I value your active participation in class, and I expect you to show evidence of being fully tuned in during class sessions.

## Assignments and projects

You can expect an assignment or project on a rough schedule of every other week. Their nature and point values will vary, although projects will tend to take more time and be worth more points than assignments.

The due date for these will be at the beginning of class on the scheduled day. For each 24-hour period thereafter, I will deduct 10% of the points possible. Note that this means that if you show up a few minutes after class starts with your assignment, you can incur the full 24-hour late penalty. I will be happy to accept solutions prior to their due date.

## Quizzes and tests

Tests will be on alternate Wednesdays, as in the tentative schedule below. Quizzes will typically be 25 minutes in length, and the midterm will be 50 minutes.

If you miss a test, you must receive advance permission from me to receive more than a 0. (Naturally dire medical emergencies usually constitute an exception.) If you are excused from the test, I will either double your lowest quiz

or exam score or administer a make-up, at my discretion. Let me know well in advance — 24 hours for exams and quizzes, and two weeks for the final. I would like to remind you that, when e-mail is impossible, telephones exist also. *Do not skip a test without my prior approval!*

Note that I may require you to document your reason for absence. Travel arrangements and work schedules are not adequate reasons to miss a test.

## Cheating and groupwork

You must properly attribute any work or ideas you use in assignments for this course which are quoted or derived from others. Plagiarism includes not only copying the ideas and the written and spoken words of others, but also copying or otherwise appropriating their computer files as well. Interfering with the work of others is also a serious academic offense. I will abide by the catalog's Academic Honesty policy in referring cases to the college's Committee on Academic Integrity.

Note that discussing or viewing others' solutions to assignments (prior to yourself handing them in) is out of bounds. Also, discussing or showing your own solution to others is wrong. A strong correlation between your solution and a classmate's solution constitutes evidence of cheating.

Some assignments will encourage you to work with another student. Unless the assignment says otherwise, this means that you and the student should submit a single solution to the assignment with both names on it.

## Tentative schedule

The schedule below is an outline of the current semester plan. Except for the test dates, it is only a starting point. Topics will be moved and even removed, and new topics may arise, in the course of the class.

Wed 25 Aug	overview, Ada control
Thu 27 Aug	Ada types
Mon 30 Aug	Ada subroutines, parameters
Wed 1 Sep	syntax, grammar, and ambiguity
Fri 3 Sep	recursive-descent parsing
Wed 8 Sep	correctness: straight-line programs
Fri 10 Sep	correctness: iterative programs
Mon 13 Sep	correctness: practice
Wed 15 Sep	<b>Quiz 1</b> , binding time
Fri 17 Sep	generics
Mon 20 Sep	stack management
Wed 22 Sep	garbage collection
Fri 24 Sep	optimization

Mon 27 Sep	Smalltalk introduction
Wed 29 Sep	OOPL compilation
Fri 1 Oct	OOPL compilation II
Mon 4 Oct	multiple inheritance
Wed 6 Oct	<b>Midterm</b>
Fri 8 Oct	lambda calculus: introduction
Mon 11 Oct	lambda calculus: recursion
Wed 13 Oct	lambda calculus: numerals
Mon 18 Oct	Haskell introduction
Wed 20 Oct	Haskell patterns
Fri 22 Oct	lazy v eager evaluation
Mon 25 Oct	Haskell types and classes
Wed 27 Oct	<b>Quiz 2</b> , Haskell structures and showing
Fri 29 Oct	Haskell printing and parsing
Mon 1 Nov	persistent trees
Wed 3 Nov	persistent queues
Fri 5 Nov	random-access lists
Mon 8 Nov	monads
Wed 10 Nov	Haskell I/O
Fri 12 Nov	functional language optimization
Mon 15 Nov	functional applications to OOP
Wed 17 Nov	<b>Quiz 3</b> , Scheme overview
Fri 19 Nov	concurrency — Ada I
Mon 22 Nov	concurrency — Ada II
Mon 29 Nov	concurrency — Java I
Wed 1 Dec	concurrency — Java II
Fri 3 Dec	concurrency — Java III
Mon 6 Dec	survey of languages
8:30a, 10 Dec	<b>Final</b>