CS 201: Fundamental Structures of Computer Science I
Fall 2006

Instructors:

Çiğdem Gündüz Demir (*Sections 1 & 2*)
EA 407A (Engineering Building), x3443
gunduz@cs.bilkent.edu.tr, demir@cs.bilkent.edu.tr
Office hours: Mon 12:40-14:30, Tue 14:40-16:30

Selim Aksoy (*Sections 3 & 4*)
EA 423 (Engineering Building), x3405
saksoy@cs.bilkent.edu.tr
Office hours: Tue 13:40-15:30, Fri 10:40-12:30

TAs:

Süleyman Tuncer Erdoğan (*Section 1*)
Umut Tosun (*Section 2*)
Hüseyin Gökhan Akçay (*Section 3*)
Emel Doğrusöz (*Section 4*)

Lectures:

Mon 10:40-12:30, Thu 8:40-9:30, BZ05 (*Section 1*)
Mon 15:40-17:30, Thu 13:40-14:30, BZ05 (*Section 2*)
Tue 15:40-17:30, Fri 13:40-14:30, BZ05 (*Section 3*)
Tue 10:40-12:30, Fri 8:40-9:30, BZ05 (*Section 4*)

Course websites: http://www.cs.bilkent.edu.tr/~saksoy/courses/cs201 (*all sections*)

Text books:

(Required) H.M. Deitel and P.J. Deitel, *C++ How to Program,* 5th ed, Prentice Hall, 2005.
(Required) F.M. Carrano, *Data Abstraction and Problem Solving with C++*, 4th ed, Addison-Wesley, 2005.
(Recommended) M.A. Weiss, *Data Structures & Algorithm Analysis in C++*, 3rd ed, Addison-Wesley, 2006.

Course Emphasis and Goals

This course introduces the abstract data types of lists, stacks and queues, and shows how one can implement them in C++ using fundamental data structures. It also introduces recursion and studies recursion as a problem solving technique using data abstraction.

Grading Policy

Quizzes: 5 %
Homework: 25 %
Midterm: 30 % (open book)
Final: 40 % (open book)

Quizzes

There will be at least two in-class quizzes. Quizzes will be given in class, without advance notice. The quizzes are going to be about the content of the lecture that is being delivered at the time of the quiz. Students that take a quiz will automatically be granted half of its full points.

Homework Assignments and Late Policy

Homework assignments will be posted on the course web site about two weeks before their due dates. Assignments are expected to be turned in by 17:00 on the due date. For the late assignments, each student will be given a total of three grace days (whole or partial) for the whole semester. Once these late days have been exhausted, no late assignments will be accepted. As an example, if Student A submits her/his 2nd assignment 29 hours late, s/he will have used two late days and have only one day left. If Students A then submits her/his 4th assignment 5 hours late, s/he will have used her/his remaining late day. If Student A submits her/his 5th assignment 1 minute late, this assignment will not be accepted.

You should turn in your homework as a hardcopy (printer output) and also upload it using the online submission form (both before the deadline). Your submission must include both source code and sample outputs; otherwise, it will not be graded. Always make sure that the code you submit does compile and run correctly. Sample outputs should test a sufficient number of inputs so as to demonstrate that your program works correctly. Graded assignments are to be picked up from the TAs during their office hours.

Late submissions should be done through e-mailing them to your instructor or TA, they should also include both source code and sample outputs. You should also turn in the hardcopy of your submission to your instructor or TA before the late days that you will be using being exhausted.

Academic Integrity

Copying or communicating during an exam is cheating. Students caught cheating on an exam will be subject to disciplinary action, as explained in the "Student Disciplinary Rules and Regulation" (www.provost.bilkent.edu.tr/procedures/AcademicHonesty.htm).

Academic integrity is a problem on programming assignments. Students naturally want to work together, and it is clear that they learn a great deal by doing so. Getting help is often the best way to interpret error messages and find bugs, even for experienced programmers. In response to this, the following rules will be in force for programming assignments:

- Students are allowed to work together in designing algorithms, in interpreting error messages, in discussing strategies for finding bugs, but NOT in writing code.
- Students may not share code, they may not copy code, and they may not discuss code in detail (line-by-line or loop-by-loop) while it is being written or afterwards. This extends up to three days after the submission deadline.
- Similarly, students may not receive detailed help on their code from individuals outside the course.
- Students may not show their code to other students as a means of helping them. Sometimes very good students who felt sorry for struggling students have provided them with "just a peek" at their code. Such "peeks" often turn into extensive copying, despite prior claims of good intentions.
- Students may not leave their code (either the electronic versions or the printed copies) in publicly accessible areas. Students may not share computers in any way when there is an assignment pending.

We use an automatic code comparison tool to help spot assignments that have been submitted in violation of these rules. The tool takes all assignments from all sections and compares them, producing a rank-ordered list of pairs of programs that are most similar. It produces a web page for each pair, highlighting the regions of the code that are similar. We check these similar pairs of assignments very carefully ourselves, and make our own judgment about which students violated the rules of academic integrity on programming assignments. *Students caught cheating on programming assignments will be subject to disciplinary action.*

Schedule

Week of	Subject	Reading	Assignments
1 Sep 11	Introduction, classes and objects, control statements	Ch.3, Ch.4, Ch.5 (Deitel)	
2 Sep 18	Functions	Ch.6 (Deitel)	
3 Sep 25	Arrays, vectors, pointers	Ch.7, Ch.8 (Deitel)	
4 Oct 2	Pointers	Ch.8 (Deitel)	HW1 out (Oct 6)
5 Oct 9	Classes: a deeper look	Ch.9, Ch.10 (Deitel)	
6 Oct 16	Operator overloading, recursion	Ch.11 (Deitel), Ch.2 (Carrano)	HW1 in (Oct 20)
7 Oct 23	Recursion	Ch.2 (Carrano)	HW2 out (Oct 27)
8 Oct 30	Algorithm analysis	Handout will be available	
9 Nov 6	Data abstraction, static lists	Ch.3 (Carrano)	HW2 in (Nov 10)
10 Nov 13	Static lists, linked lists	Ch.3, Ch. 4 (Carrano)	HW3 out (Nov 17)
11 Nov 20	Linked lists	Ch.4 (Carrano)	Midterm (Nov 25)
12 Nov 27	Stacks	Ch.6 (Carrano)	HW3 in (Dec 1) HW4 out (Dec 1)
13 Dec 4	Stacks	Ch.6, Ch.5.2 (Carrano)	
14 Dec 11	Queues	Ch.7 (Carrano)	HW4 in (Dec 15) HW5 out (Dec 15)
15 Dec 18	Queues	Ch.7 (Carrano)	
			HW5 in (Dec 29)