Computer Science & Engineering 143
Computer Programming II

Instructor	Benson Limketkai
email	bensonl@cs.washington.edu
phone	616-1246
office	Paul G. Allen Center, room 210
office hours	*see course web page*

Lecture and Section Times

Lecture	MWF 10:50—11:50, EEB 105
Sections	various times and locations on Tuesdays and Thursdays

Textbook

Building Java Programs, 2nd edition, Reges & Stepp, required (although students who purchased the first edition when they took CSE142 are not expected to purchase the second edition).

Course Overview

This course is a continuation of CSE142. While CSE142 focused on control issues (loops, conditionals, methods, parameter passing, etc), CSE143 focuses on data issues. Topics include: ADTs (abstract data types), stacks, queues, linked lists, binary trees, recursion, interfaces, inheritance and encapsulation. The course also introduces the notion of complexity and performance tradeoffs in examining classic algorithms such as sorting and searching and classic data structures such as lists, sets and maps. The course will include a mixture of data structure implementation as well as using off-the-shelf components from the Java Collections Framework. The prerequisite is CSE142 or equivalent.

Discussion Sections

You are encouraged to participate in two weekly 50-minute discussion sections. The TA who runs your discussion section will grade your homework assignments. In section we will answer questions, go over common errors in homework solutions and discuss sample problems in more detail than we can in lecture.

Grading

You will be expected to complete a variety of programming assignments for this course and to take two open-note, open-book exams. The resulting scores will be combined according to the following weightings:

40%	weekly homework assignments
20%	midterm (in class on Monday, 7/26/10)
40%	final exam (in class on Wednesday, 8/18/10, and Friday, 8/20/10)

Using the weightings above, each student's scores will be turned into an overall score ranging from 0 to 100 percent. These will be turned into grades as follows:

90%	at least 3.5
80%	at least 2.5
70%	at least 1.5
60%	at least 0.7

If you need to miss an exam, you must contact the instructor *prior* to the exam to get permission. Even if you are sick at home, you should be able to call your instructor's office phone number to leave a message that you need to be contacted.

The weekly assignments will generally be graded on a 20-point scale.

Course Administrator

Pim Lustig (pl@cs.washington.edu, 616-3225) is the course administrator and will handle many details including registration and switching sections.

Course Web Page

Information about the course will be kept at http://www.cs.washington.edu/143. Links to course handouts will be kept on this page along with useful links to other class resources.

Computer Access/Software

The department operates an Introductory Programming Lab (IPL) that is located on the third floor of Mary Gates Hall (MGH 334). TAs will be available at the lab to help students with problems. You can use any Java environment you want although the recommended software for this course is the Java Development Kit (JDK) version 6 and the jGRASP editor. More information can be found on the class web page under the "Working at Home" link.

Late Policy

Each assignment will list its due date. Assignments will be due on Thursdays at 9 pm. Each student in the class will have a total of three "free" late days (a late day is 24 hours of lateness). There are no partial days, so assignments are either on time, 1 day late, 2 days late, etc. Because of this generous late policy, students will not be granted extensions for assignments unless they have some highly extenuating circumstances. Once a student has used up all of his or her late days, each successive late day will result in a loss of 1 point. No assignment will be accepted more than 3 days after its due-date. No assignment can be submitted after 9 pm of the last day of class (Friday, August 20th), whether or not a student has free late days left.

Policy on Collaboration

You are to complete programming assignments individually. You may discuss the assignment in general terms with other students including a discussion of how to approach the problem, but the code you write must be your own. The intent is to allow you to get some help when you are stuck, but this help should be limited and should never involve details of how to code a solution. **You must abide by the following:**

- You may **not** work as a partner with another student on an assignment.
- You may **not** show another student your solution to an assignment.
- You may **not** have another person (current student, former student, tutor, friend, anyone) "walk you through" how to solve an assignment.

Under our policy, a student who gives inappropriate help is equally guilty with one who receives it. Instead of providing such help, refer other students to class resources (lecture examples, the textbook, the IPL, or emailing a TA or instructor). You must not share your solution and ideas with others. You must also ensure that your work is not copied by others by not leaving it in public places, emailing it others, posting it on the web, etc.

If you are taking the course a second time, you are allowed to submit a previous solution that you authored unless that program was involved in a case of academic misconduct. For any assignment where academic misconduct was found (whether the case was settled formally or informally), you have to write a new version of the program.

We enforce this policy by running similarity-detection software over all submitted student programs, including programs from past quarters.