CS230 Data Structures **Handout # 1**
Prof. Lyn Turbak **January 29, 2007**
Wellesley College

CS230 Course Information

1 Contact Information

Lecturer:

Name:	Franklyn Turbak (please call me "Lyn")
Office:	SCI E126
Phone:	x3049
E-mail:	`fturbak@wellesley.edu` ("Franklyn Turbak" in FirstClass)
Lectures:	SCI E111, Mon./Thu. 9:50-11am
Office Hours:	Monday: 4pm–6pm Tuesday: 12:30-2:30pm Wednesday: 4–6pm Friday: 3–5pm Appointments can be made for other times. This semester, I will be off-campus most Thursday afternoons, so I will almost never be available on Thursday afternoons. I will sometimes need to cancel or shift office hours to attend a meeting or at talk. I will post a message to `CS230-S07 Announcements` to announce changes to office hours.

Lab Instructor:

Name:	Stella Kakavouli (please call me "Stella")
Office:	SCI E131 (in the micro-focus)
Phone:	x3120
E-mail:	`skakavou@wellesley.edu` ("Stella Kakavouli" in FirstClass)
Labs:	Lab 01: Tue. 1:30–3:20pm; Lab 02: Tue. 6:00–7:50pm. *All labs are held in the Linux lab (a.k.a. the "micro–focus")*
Office Hours:	Monday: 11am–12:30pm Tuesday: 4:00–5:30pm Thursday: 11am–noon Appointments can be made for other times.

Web Site: `http://cs.wellesley.edu/~cs230`

First Class: `CS230-S07 Announcements`, `CS230-S07 Q&A`

Drop-in Tutors: Amy Dai, drop-in hours Sundays 6-8pm
Cassie McLeod, drop-in hours Mondays 7-9pm

2 Course Overview

CS230 focuses on five "big ideas":

Data Abstraction: In order to cope with the complexity of large writing programs, it is desirable to express them as combinations of components whose behavior can be understood independently from their implementation. In CS111, you learned how methods provide an abstraction barrier between the caller and implementer of a method. We will review this notion and introduce abstract data types (ADTs) as a way of thinking about computational values and the operators that manipulate them.

Modularity: The ability to create complex artifacts (computational or otherwise) is enhanced if they can be composed out of reusable components with standard interfaces that can be combined in mix-and-match ways. We will study how to compose programs out of such components.

Performance Analysis: Programs are often judged by how effectively they make use of resources such as space and time. We will study ways to describe the efficiency of algorithms and use these to evaluate various approaches to implementing data structures and algorithms.

Standard Data Structures: We will learn some classical data structures and algorithms that should be in every programmer's bag of tricks. The data structures include arrays, vectors, enumerations, lists, trees, stacks, queues, priority queues, sets, bags, tables and graphs.

Program Development: Unlike in CS111, where we mostly modified exisiting programs, in CS230 we will be creating many programs from scratch. We will get lots of practice with designing, implementing, testing, and debugging programs.

We will use Java as the language for exploring these ideas.

3 Prerequisites

The prerequisite for CS230 is CS111, *Computer Programming and Problem Solving*. Students with significant programming experience who have not taken CS111 may take the course with permission of the instructor.

4 Classes

There are two 70-minute lectures each week, which will introduce the main content of the course. Every student is also required to attend one 110-minute lab each week. Lab work will include exercises to review and reinforce the lecture material and to develop general programming, testing and debugging skills. The labs will also provide further opportunity to ask questions about course material and discuss homework assignments.

5 Reading Materials

We will not be using a textbook in CS230 this semester. You should plan on taking good notes during lecture. Supplementary notes will occassionally be handed out.

Although there is no required textbook, you are encouraged to consult textbooks on Java and/or data structures on a frequent basis. A valuable on-line resource is Allen Downey's *How to Think Like a Computer Scientist: Java Version*, which can be found at

`http://www.ibiblio.org/obp/thinkCSjav/`

and is also linked from the *Documentation* page of the CS230 web site. A large part of this text provides a review of basic Java, which we will move through quickly during the first few weeks of the semester. The new material for CS230 is covered in Chapters 14-19, which will be the focus of the rest of the course. There will be suggested readings from this book.

The syllabus includes *optional* readings from two other books:

1. Frank Carrano, *Data Structures and Abstractions with Java*, Pearson/Prentice Hall, 2007 (2nd edition) (The first edition, Frank M. Carrano and Janet J. Prichard's *Data Abstraction and Problem Solving with Java: Walls and Mirrors* is also helpful.)

2. Sesh Venugopoal, *Data Structures Outside In with Java*, Pearson/Prentice Hall, 2007

Copies of these two books, as well as many other textbooks on data structures in Java, can be found in the CS111 student lounge in SCI 173. This room is open 24/7. You are encouraged to consult the books in 173 whenever you feel you need additional information about a topic from class. Please do not remove the books from 173/micro-focus area so that they remain available for all students. Below is a list of some of the other books available in the SCI 173 library that are especially relevant to CS230:

- Duane Bailey, *Java Structures*, McGraw Hill, 1999.
- Timothy Budd, *Classic Data Structures in Java*, Addison Wesley, 2001.
- William J. Collins, *Data Structures and the Java Collections Framework*, McGraw Hill, 2002.
- Michael T. Goodrich and Roberto Tomassia, *Data Structures and Algorithms in Java*, John Wiley & Sons, Inc., 2001 (2nd edition).
- John Lewis & Joseph Chase, *Java: Software Structures*, Pearson/Addison-Wesley, 2004.
- Michael Main, *Data Structures & Other Objects Using Java*, Addison-Wesley, 1999.
- David D. Riley, *The Object of Data Abstraction and Structures using Java*, Addison-Wesley, 2003.
- Glenn W. Rowe, *An Introduction to Data Structures and Algorithms with Java*, Prentice-Hall Europe, 1998.
- Sartaj Sahni, *Data Structures, Algorithms, and Applications in Java*, McGraw-Hill, 2000.
- Thomas Standish, *Data Structures in Java*, Addison-Wesley, 1998.
- Mark Allen Weiss, *Data Structures & Problem Solving Using Java*, Addison-Wesley, 1998 (1st edition) 2002 (2nd edition)

6 Course Web Pages, Directories, and Conferences

All handouts and various course-related links can be found on the CS230 home page:

`http://cs.wellesley.edu/~cs230`

The CS230 course directory is located on `cs.wellesley.edu` in the directory `/home/cs230`. This directory contains material relevant to the class, and is where the problem set drop folders are located. The CS230 directory can be accessed by connecting to `cs.wellesley.edu` and navigating to `/home/cs230` using `scp` in Linux, WinSCP on a PC, or Fetch on a Mac,

Additionally, there is a `CS230-S07` conference in FirstClass with two subconferences:

- `CS230-S07 Announcements` will be used to make class announcements, such as corrections to assignments and clarifications of material discussed in class.
- `CS230-S07 Q&A` is a forum for you to post questions or comments. They will be answered by me or a classmate. This is also a good place to find people to form a study group.

You should plan on reading the CS230 conferences on a regular basis. It is strongly recommended that you add both subconferences to your FirstClass desktop

7 Assignments

7.1 Problem Sets

There will be seven problem sets in which you will write Java programs or answer pencil-and-paper problems that illustrate concepts discussed in class. Many of the assignments will be challenging. Keep in mind that programming often consumes more time than you think it will. Start your assignments early! This will give you time to think about the problems and ask questions if you hit an impasse. Waiting until the last minute to begin an assignment is a recipe for disaster.

Appropriate documentation, good programming style and comprehensive testing are all essential parts that will be given equal weight to program correctness. See Takis Metaxas's *On Good Programming Style* (available from the *Documentation* page of the CS230 web site) for some guidelines on programming style.

Unless otherwise announced, problem sets are due at the time listed on the top of the problem set handout. Problem sets will typically be due on Tuesdays before Spring Break, and on Thursdays after Spring Break. You should turn in both a "hard" (paper) copy of your assignment and a "soft" (electronic) copy of any programs from the assignment. Soft copies should be submitted to your drop folder for assignments, which can be found at:

`cs.wellesley.edu:/home/cs230/drop/`*assignment-number*`/`*your-username*.

The softcopy submission will typically be a folder containing your programs and any other information you think is appropriate. The problem set handouts will provide additional instructions on how to submit your softcopy.

Assignments will be graded on a 100 point scale. We will try to have assignments graded as soon as possible. Solutions to assignments will be posted on the web.

7.2 Collaboration Policy

We believe that collaboration fosters a healthy and enjoyable educational environment. For this reason, we encourage you to talk with other students about the course and to form study groups.

Because the programming assignments in this course are challenging, you will be allowed on any assignment to form a two-person "team" with a partner. The two team members can (in fact,

must; see below) work closely together on the assignment and turn in a single hard- and soft-copy of the assignment for the team. The grade received on such a submission will be given to both team members.

This is a rather unusual collaboration policy, and it is only allowed subject to the following ground rules:

- The work must be a true collaboration in which each member of the team will carry her own weight. It is *not* acceptable for two team members to split the problems of the assignment between them and work on them independently. Instead, the two team members must actively work together on all parts of the assignment. In particular, almost all programming on the assignment should be done with the two team members working at the same computer. It is strongly recommended that both team members share the responsibility of "driving" (i.e., typing at the keyboard), swapping every so often.

  The fact that team members have to program together means that you need to carefully consider a potential partner's schedule before forming a team. You cannot be a team if you cannot find large chunks of time to spend at a computer together!

- You can only work with a given partner on a single problem set during the semester. So if you want to continue to collaborate, you must choose a different partner for every assignment. Rotating through partners is a good way to build community in the class and is helpful for avoiding situations where one individual feels pressured to continue working with another. (If you have exhausted all possible partners with a similar working schedule but still wish to collaborate on remaining assignments, contact Lyn.)

- You are not *required* to have a partner on any assignment, but you are *encouraged* to do so. Based on past experience, working with a partner can significantly decrease the amount of time you spend on an assignment, because you are more likely to avoid silly errors and blind alleys. On the other hand, it can be difficult for teammates to find time to work together, and certain individuals may take more time on an assignment than they would alone. In this case there are still benefits to working with a partner, but they may be outweighed by the time cost.

Unless otherwise instructed, teams are allowed to discuss problem sets with other teams and exchange ideas about how to solve them. However, there is a thin line between collaboration and plagiarizing the work of others. Therefore, we require that each (one-person or two-person) team must compose its own solution to each assignment. In particular, while you may discuss strategies for approaching the programming assignments with other teams and may receive debugging help from them, each team is required to write all of its own code. **It is *unacceptable* (1) to write a program with another team and turn in two copies of the same program or (2) to copy code written by other teams or students from previous semesters; (3) to copy any solution code written by the instructors from this semester or previous semesters; or (4) to give or receive help from other teams involving debugging a particular Java program[1]. We consider any of the actions listed above to be violations of the Honor Code that we will report to the General Judiciary.**

In keeping with the standards of the scientific community, **you must give credit where credit is due.** If you make use of an idea that was developed by (or jointly with) others, please reference them appropriately in your work. E.g., if person X gets a key idea for solving a problem from person Y, person X's solution should begin with a note that says "I worked with Y on

[1]It's OK to ask fellow students what the likely cause of particular error message is. And it's OK to receive help from the instructors at any time. What's not OK is to have a member of another team peering at your code over your shoulder helping you to debug it.

this problem" and should say "The main idea (due to Y) is ..." in the appropriate places. It is unacceptable for students to work together but not to acknowledge each other in their write-ups.

When working on homework problems, it is perfectly reasonable to use code from the textbooks and other materials handed out in class. It is also reasonable to consult public literature (books, articles, etc.) for hints, techniques, and even solutions. However, you must cite any sources that contribute to your solution. There is one extremely important exception to this policy: **assignments and solutions from previous terms of CS230 are *not* considered to be part of the "public" literature. You must refrain from looking at any solutions to problem sets or exams previous semesters of CS230. It is our policy that consulting solutions from previous terms of CS230 constitutes a violation of the Honor Code.**

Despite the existence of the Honor Code, we have had trouble with students following the above policies in the past. To remind you of these policies, we will ask you to sign each problem set and exam header sheet saying that you have followed the policies expressed above. Keep in mind that violating the policies can lead to a serious penalty. Violators from previous semesters have asked us to sternly warn you that if the General Judiciary finds you guilty of violating the policy on even a single problem set or exam, it can lead to a zero score for the *entire* problem set component or exam component of your grade, which typically results in a **failing grade** in the course.

7.3 Late Homework Policy

We realize that it is not always possible to turn in problem sets on time. On the other hand, turning in one problem set late can make it more difficult to turn in the next problem set on time. We have decided on the following policy for this course this term:

> All problem sets will be due by the announced time on the due day. A problem set can be turned in $24 \cdot n$ hours late if it is accompanied by n Lateness Coupons. If you work with a partner, each of you needs to attach one Lateness Coupon per person per day late.

At the end of this handout, you will find *seven* Lateness Coupons that you can use throughout the term. Use them wisely: you only get seven, and they are not copyable or transferable between students. (You also cannot use them on exams!)

You may turn in late problem sets by slipping them under my office door. Of course, if we post solutions before you turn in a late problem set, you are bound by the Honor Code not to examine these solutions.

In extenuating circumstances (e.g., sickness, personal crisis, family problems), you may request an extension without penalty. Such extensions are more likely to be granted if they are made before the due date.

7.4 Problem Set Header Sheets

We would like to get a sense for how much time it takes you to do your CS230 problem sets. Please keep track of the time you spend on each problem of your problem sets, and include this information on the problem set header sheets that will be provided for each problem set. Turn in this header sheet as the first page of your hardcopy solutions.

7.5 Final Project

During the last month of the semester, you will work (alone, or with a partner) on an extended, individualized programming project that you will design and build from scratch. After choosing an interesting application or problem, you will first build a skeleton of the object classes, methods,

abstract data types and user interface needed for your application, and then fill in the details to create a fully working implementation. Important deadlines in the final project are as follows:

1. Thu. Mar. 29: Project Phase 1: Project Proposal
2. Thu. Apr. 18: Project Phase 2: Program Outline
3. Mon. May 7: Project Phase 3: Detailed Program Skeleton
4. Tue. May 8: Project Phase 4: Informal Presentation (in lab)
5. Mon. May 14: Project Phase 5: Code Review
6. Mon. May 21: Project Phase 6: Demonstration and Final Program (can be submitted sooner by those leaving earlier for the summer)

More details on the project and its individual phases will be posted later in the semester.

7.6 Extra Credit

This semester we do not plan to give any "official" extra credit problems. But if we am impressed by work that you have done on challenging problems related to the course, we may give extra credit points for such work.

7.7 Programming

All programming in this course will be done using Java on the Linux platform. You will edit programs using the Emacs editor. Details on Linux, Emacs, and using Java in Linux are provided in Handout #3.

In order to use the Linux workstations, you will need a Linux account. If you took CS111, we will upgrade your account so that is "powerful" enough to run Linux. If you did not take CS111, you need to request a new account by filling out the form at:

`http://cs.wellesley.edu/accounts/account-request.html`

The "default" place for you to work on your assignments will be at the CS Department's Linux workstations in the micro-focus. If you desire, you are welcome to access the micro-focus machines remotely[2] or to install Linux on your own computer so that you can use it for your CS230 work. However, many students prefer to work directly at the Linux workstations in the micro-focus since there are likely to be other CS230 students working there, increasing the probability of collaboration.

For many Java programming problems in this course, you will find it helpful to consult the Java 5.0 API documentation at:

`http://java.sun.com/j2se/1.5.0/docs/api/`

This URL is linked from the *Documentation* section of the CS230 web site.

7.8 Saving Work

You have a limited amount of space on the CS department fileserver (`cs.wellesley.edu`) to store your course-related files.

Make backups (onto a flash drive, your FirstClass account, or the hard disk of your own computer) of all your work throughout the semester. In the past, numerous students have lost all of

[2]You can connect to any CS Dept. Linux computer remotely using an SSH client like `ssh` on a Mac or `PuTTY` or `TTSSH` on a PC. When connected via SSH, you can even display X Windows from a Linux computer by using the X11 server on a Mac or the free Xming server on a PC. Consult Lyn for details.

their work when they could least afford it. Don't join them! For example, in March 2004, many people lost a week's worth of work when a hacker deleted much of the puma filesystem.

While planning your computer usage, keep in mind that computers do break down and Wellesley's are no exception.

8 Exams

There will be two CS230 exams, both of which are open book and open notes:

- A one-week long take-home exam that will be posted on Tue. Mar. 6, and due on Tue. Mar. 13.
- A in-class exam in lab on Tue. May 1.

Please mark these dates in your calendars. If you have any conflicts regarding the exam dates, you must contact your instructor as soon as possible.

The take-home exam will be similar to a problem set and will require use of a computer. **You are not allowed to collaborate with anyone else or receive any help from the instructors on any of the exams.** Any deviation from this policy is a serious Honor Code violation.

9 Grades

The course grade will be computed as shown below:

Problem sets (total)	40%
Final project	20%
Exam 1	20%
Exam 2	15%
Class participation	5%

All assignments and exams count toward the final grade. Class participation includes attendance in lectures and labs, as well as active participation in class. Participation is evaluated on a five-point scale (1: rarely participates; 5: frequently participates) and can make a difference in your final grade if you are near the border between two grade levels.

The default ranges for grades are expressed as a percentage of total points:

A	93.33 – 100
A-	90 – 93.32
B+	86.66 – 89.99
B	83.33 – 86.65
B-	80 – 83.32
C+	76.66-79.99
C	73.33-76.65
C-	70 – 73.32
D	60 – 69.99
F	below 60

We reserve the right to lower boundaries between grades, but we will not raise them. This means that we can grade on a curve, but only in your favor.

The above information is intended to tell you how we grade. It is not intended to instill in you a preoccupation with point accumulation. We encourage you to treat points in this class as you would Monopoly money. If you focus on learning the material, the grade will take care of itself.

10 Finding Help

If you have any questions at all about the class (whether big or small, whether on problem sets lectures, reading, or whatever) please contact Lyn or Stella. **That's what we're here for!**

The best time to see us is during our scheduled office hours (which are listed at the top of this handout). If these times are not convenient, we can set up an appointment at some other time. You can set up an appointment by talking with us in person, calling us on the phone, or sending us email.

This semester we are fortunate to have two drop-in tutors: Amy Dai and Cassie McLeod. They will hold drop-in hours in the micro-focus during the hours posted on the CS230 home page.

If you are having trouble with the course, you can request a one-on-one tutor from the Learning and Teaching Center (LTC). This service is confidential and free of charge; please take advantage of it if you would like some extra help! Contact the instructors or LTC for more information about this service.

Finally, when looking for help, don't overlook your fellow students — not only those who have taken the course in the past, but your classmates as well. Your classmates are a valuable resource; make good use of them!

11 Students With Special Needs

If you have any disabillities (including "hidden" ones, like learning disabilities), we encourage you to meet with us so that we can discuss accomodations that may be helpful to you.

12 Mathematical Modeling Distribution

CS230 counts for one Mathematical Modeling (MM) distribution credit. It does *not* satisfy the laboratory requirement or QR overlay requirement.

LATENESS COUPONS

Below are seven Lateness Coupons. A problem set that is $24 \cdot n$ hours late must be accompanied with n Lateness Coupons in order to be accepted. That is, each coupon gives you one extra day to turn in a problem set. You may use them in any manner in which you wish – e.g., turn in every problem set one day late, or turn in one problem set seven days late. Lateness coupons are not transferable between students, and may not be used on exams. If a two-person team turns in an assignment late, each team member must use the appropriate number of coupons.

CS230 Lateness Coupon #1
CS230 Lateness Coupon #2
CS230 Lateness Coupon #3
CS230 Lateness Coupon #4
CS230 Lateness Coupon #5
CS230 Lateness Coupon #6
CS230 Lateness Coupon #7