CSCI 1902: Structure of Computer Programming II

6/12-8/4/2006 ; lecture MW 9:05-11:45, CS/EE 3-115 ; lab Th 9:00-10:30 CS/EE 1-260 ; 4 cr.

1 Teaching Staff

Instructor:	**Reid Priedhorsky** (`reid@umn.edu`)
Teaching Assistant:	**Andy Exley** (`exley@cs.umn.edu`)

See the course website for information on office hours.

2 About the Course

2.1 Course Description

CSCI 1902 introduces basic data structures and their related algorithms, in particular linked lists, queues, stacks, and trees. These are among the most fundamental building blocks of computer science, and nearly all nontrivial programs make use of them, whether explicitly or implicitly. We will approach this material in an object-oriented framework, so concepts like class, object, method, and inheritance will also be covered.

CSCI 1902 is a required course for computer science majors, and it is a prerequisite for most of the higher-level computer science courses. The programming language used for this course is Java.

2.2 Goals and Objectives

This course is taught for mastery and retention. Mastery and retention means that, at some unspecified time in the future (future courses, the workplace, etc.), you will be able to use the techniques, skills, and knowlege taught in this course to (a) solve novel problems, (b) justify and explain your reasoning, and (c) intelligently critique the reasoning of others. It is not necessary to have all details at immediate recall, but successful students will have mastered the material to a great enough depth that external sources are consulted for details and reminders only, not "big picture" concepts or re-learning.

Upon successful completion of this course, you will be able to:

- Solve moderately complex real-world problems using the Java language. For example, you will be able to compute the volume and other physical properties of a complex geometrical object which consists of a hierarchical collection of varied subobjects.
- Learn, on your own, an arbitrary object-oriented programming language (e.g. Python), and then use it to the same effectiveness as Java.
- Select and implement, for a given application, the most appropriate data structures and algorithms, from those presented, in an object-oriented fashion. For example, you will be able to select and implement sorting algorithms for large, very large, medium, and small data sets under various constraints of time and space.
- Clearly and effectively explain and justify your choices of selection and implementation.

- Evaluate and critique the selections, implementations, and justifications of others.
- Effectively debug your work and verify its performance and correctness.

CSCI 1902 will also exercise and further develop your general analytical thinking and problem solving skills.

2.3 Prerequisites

CSCI 1901 is the only formal prerequisite for CSCI 1902. Students should also have experience in analytical thinking and problem solving and experience in using common computer science methods in problem solving.

2.4 Textbook

Frank M. Carrano and Walter Savitch, *Data Structures and Abstractions with Java*, Pearson Education, 2003.

2.5 Computer Labs

We will target Unix ITLabs machines for the programming assignments; there are various ITLabs throughout campus containing such machines available for your use. (See the course FAQ for more information.) You must have an ITLabs account in order to submit lab assignments.

Java is freely available for a variety of machines, and you may do your work on the system of your choice. However, your work will be graded on, and must function correctly on, ITLabs UNIX machines. If you use only standard Java features, you should have no problems, but it is **your responsibility** to test that this is indeed the case. **Note that we cannot provide help or support for environments other than ITLabs SunOS, Solaris, or Linux.**

2.6 On-Line Course Information

The course web page, at `http://www.itlabs.umn.edu/classes/Summer-2006/csci1902`, contains announcements, notes, this syllabus, a forum, and many other resources for your use. This web site will be the central clearinghouse for information about the course. **We expect you to be familiar with all parts of the course website** and take advantage of all the resources we make available to you there. In particular, you should read any new announcements and skim any useful forum postings at least weekly (and more frequently during busy times).

The forum is a highly useful tool and builds community among the students. We very strongly encourage you to actively participate.

2.7 Course Format

Weekly class meetings for this course consist of five hours of lecture and one and a half hours of lab; the course runs eight weeks.

The primary teaching method in this course is lecture, delivered in short mini-lectures of length appropriate for the complexity of the material and the attention span of you, the student. I supplement this with a variety of other active learning methods, some individual, and some group-based.

I work to foster a student community in this course. Students are more motivated when they feel they are part of a greater whole, and a feeling of community encourages teaching interactions between students, which benefit the learning of all students involved. We'll use three tools to this end: the online discussion forum, assigned groups, and self-selected groups. Assigned groups reduce the terror of forming one's own group in a class filled with strangers, and they foster interactions between students who might otherwise not interact. On the other hand, students often prefer self-selected groups. For programming assignments, you may work in self-selected groups of two, or individually, as you prefer. In class, you will frequently do work in assigned or self-selected groups.

The forum is intended to be a place for all students to gather and discuss the course and course issues. Andy and I will step in to moderate and answer questions as needed, but we encourage you to answer each others' questions. We will generally take a laissez-faire attitude towards management: posts will be removed only if we think they inappropriately reveal answers, may be illegal, or may violate University or department policies (e.g. harrassing or threatening posts). Off-topic posts will be removed only if they are beginning to overwhelm the on-topic posts. Complaining about the course and teaching staff is explicitly permitted, but do be aware that we read what you write.

Office hours are a key component of success, because they permit flexible one-on-one interaction not possible in class. I specifically invite you to visit and would be most pleased if you did so.

3 Coursework and Grading

3.1 Components of Your Grade

20%	Midterm Exam
30%	Final Exam
40%	Programming projects (five @ 8%)
10%	Exam question proposals (two @ 5%)

I encourage neatness, organization, and clarity in student work, because these are components of good communication skills, which are key to success in academia, work, and life. I award a 5% grade bonus on exceptionally neat work, and I reserve the right to refuse exceptionally messy work. ("Exceptionally messy" includes packets of paper attached together with something other than a staple: I do not accept work paperclipped together, corner folded down, etc.)

You are responsible for choosing a load (school, work, social life, etc.) that you can handle. If you overload yourself, you increase the likelihood that your grades will go down. University policy for normal courses is that students should plan to spend at least 8 hours per week outside of class and 4 hours in class for a four-credit class just to receive a C grade. Because this is a summer course, we pack 15 weeks of work into just 8, so weekly workload will increase accordingly.

For information on the meaning of letter grades, please refer to the section Grading Policy in the Undergradute Catalog[1].

3.2 Exams

There are two exams: a midterm and a final. Both are closed-book; however, you may bring one letter-size piece of paper, two-sided, of **handwritten** notes. The final is comprehensive, with approximately two-thirds of the material being post-midterm.

Exam grading is criterion-based (i.e. not curved), and exams are normed against Bloom's Taxonomy of learning:[2]

1. **Knowledge**. The student can recall data or information. For example, a student might recall the time complexities of standard sorting algorithms.

2. **Comprehension**. The student can understand the meaning, translation, and interpretation of instructions and problems. The student can state a problem in his or her own words. For example, a student might write down in pseudocode a previously learned sorting algorithm.

3. **Application**. The student can apply concepts in novel situations. For example, a student might translate a sorting algorithm from one language to another, or a student might adapt a sorting algorithm to a new list data structure.

4. **Analysis.** The student can separate material or concepts into component parts so that organizational structure can be understood. The student can distinguish between facts and inferences. For example, a student might figure the time complexity of a novel algorithm.

5. **Synthesis**. The student can build a structure or pattern from diverse elements. The student can put parts together to form a whole, with emphasis on creating a new meaning or structure. For example, a student might create a program, using previously learned techniques and tools, to meet a specification.

6. **Evaluation**. The student can make judgements about the value of ideas or materials. For example, based on needs of a particular situation, a student might select a sorting algorithm and explain his or her reasoning in the choice.

Each exam question is targeted to a specific level of learning and will be indicated as such on the exam handout. You will earn a numeric score, as is traditional, but the weights will be figured such that to earn the grade:

- C, you must answer all Knowledge, Comprehension, and Application questions substantially correctly.

- B, you must additionally answer all Analysis questions substantially correctly.

- A, you must additionally answer all Synthesis and most Evaluation questions substantially correctly.

[1] `http://www.catalogs.umn.edu/ug/gen/policies.html\#policy`

[2] This section is based on `http://www.nwlink.com/~donclark/hrd/bloom.html`.

Sample exams will be handed out, and you will have the opportunity to propose exam questions (indeed, this is required). **The 90-minute final exam will be held during the last discussion period on August 3.**

3.3 Assignments

3.3.1 Programming Projects

There will be five graded programming projects, due approximately weekly. Assignments are due at 11:59pm on the due date unless otherwise specified. Note that the details of each project are tentative and subject to change.

- **Project 0: Getting Started.** This project is ungraded. You will demonstrate that you can accomplish administrative tasks necessary for success in the course, such as opening your ITLabs account, compiling and running a Java program, using the submit tool, and making a post in the forum.
- **Project 1: Loops.** You will calculate triangular and happy numbers using loops.
- **Project 2: Sorting.** You will implement several sorting algorithms and compare their performance on a variety of inputs.
- **Project 3: Linked Lists.** You will solve the Josephus Problem, a classic combinatorics problem, using linked lists.
- **Project 4: Stacks.** You will implement a Reverse Polish Notation (RPN) calculator using a stack.
- **Project 5: Linked List vs. Search Tree.** You will implement a spelling checker, using both a linked list and a binary search tree to store the dictionary, and compare the performance of the two methods.

3.3.2 Exam Question Proposals

These will be due the class session before each exam and are intended as a learning/study aid. Due to the time pressure, no grace days may be used on exam question proposals.

For each proposal, you must write six questions, one targeting each level of Bloom's Taxonomy. Along with each question, you must write an answer you deem correct and sufficient for full credit.

Grading will be credit/no credit; these are not graded for correctness. To earn credit, your work must demonstrate that you have, in good faith, created high-quality exam questions; in other words, your work must show substantial thought and effort.

The questions you submit are candidates for inclusion on the exam. Submissions with a question selected for the exam will receive 20% extra credit, so be thoughtful. Exams will contain between 0 and 50% student-submitted questions or adaptations of them, and it is highly unlikely that we will select more than one question from any student submission. Note that these assignments are learning tools for you, not a scheme to reduce the amount of work required to create the exams.

These are individual assignments, meaning you must write the questions and answers yourself. However, we encourage you to share your end product with others to assist in studying.

4 Class Policies

4.1 Late Homework

Late homework will not be accepted. This means that there is no sliding grade penalty; late work will receive a zero. Problem sets are due at the beginning of class on the due date. In other words, do not skip class or arrive tardy because you were finishing your homework. Work is considered to have been submitted when the last portion is submitted. In particular, this means that corrections to submissions are not accepted after the due date.

There are two and only two exceptions to this policy.

- Emergencies which cannot be anticipated, e.g. illness, family emergency, etc. Please contact us as soon as possible once you know the emergency exists, and we will be happy to work with you. Documentation of the emergency will probably be required.
- "Grace days." Each student has **one** late work grace day to use at any time during the semester — you may turn in one assignment one day late. If you use a grace day, you must note this fact prominently on your turned-in work, and please contact us to arrange turning in hardcopy on non-class days. There is no grade penalty for use of your grace day.

It is always better to turn in incomplete work, earning partial credit, rather than nothing at all. If you have any concerns on this matter, please do not hesistate to contact us.

4.2 Make-Up Exams

Make-up exams will generally not be given, and unexcused absences on exam day will incur a grade of zero on the exam. In case of unforeseeable emergency, please contact us as soon as possible to arrange a make-up exam or other workaround.

4.3 Grading Problems

We are human, and we occasionally make mistakes in grading. We strongly encourage you to review our grading of your assignments and exams, and we are more than happy to hear your concerns on this issue. In particular, you should check your grades in GRIT after every assignment and before the final exam. GRIT is automatically generated from our master gradebook, so errors there imply errors in your final grade.

To be considered, grading irregularities or other problems must be brought to our attention within 10 calendar days after feedback is sent or made available. This period starts the day we first bring graded items to class, announce availability, or email feedback, *not* the day you pick up the item or read your email. Please contact Andy first with problems, but do not hesitate to contact me if you are unable to resolve the problem.

In particular, we strongly encourage you to pick up your graded final exam and review our grading. Be aware, however, that the teaching staff must submit final grades within 72 hours of the final exam time. After this, grade changes are more difficult administratively and take longer.

Please keep all returned work until final grades are posted.

4.4 Attendance

Attendance in both lecture and lab is at your option. However, if you skip class, it is not reasonable to expect the teaching staff to fill you in on what you missed. Furthermore, you are responsible for knowing what goes on during class, regardless of whether you attend or not, i.e., do not expect everything to be posted on the class website.

That said, we most emphatically encourage you to attend. We work hard to make class time valuable for students, and student learning correlates very strongly with attendance. If you are finding class time unhelpful, we would much prefer that you let us know, so we can identify the problem and improve our teaching, rather than simply not attending.

4.5 Good Faith Interpretation of Directions

You are required to make a good-faith effort to follow the spirit of any directions or instructions we give with class work; in other words, you are not permitted to "game the system" or interpret assignments in strange ways to make things easier for you.

Some examples: if you encounter errors or ambiguity in assignments, we expect you to expeditiously seek clarification from us; or, if you encounter apparent bugs in the tools or code we provide, we again expect you to seek clarification rather than concluding they are broken and proceeding another way.

5 Department, College, and University Policies

5.1 Incompletes

We will grant an incomplete very rarely, and only in instances when an unforseeable event causes a student who has completed all the coursework to date to be unable to complete a small portion of the work (typically the final assignment or exam). Incompletes will not be granted for forseeable events, including heavy course load or poorer-than-expected performance. Verifiable documentation must be provided, and arrangments for the incomplete should be made as soon as such an unforseeable event is apparent.

5.2 Withdrawals

June 25 is the last day to drop the course without a "W" grade. Through July 7, you may withdraw with a "W" grade without permission from anyone. After that, withdrawal is up to your college and is not automatic.

If you are having trouble and are considering dropping the course, I encourage you to come talk to me before doing so. While this decision is up to you, I'd like the opportunity to help you figure out a solution short of dropping.

5.3 Scholastic Conduct

Academic dishonesty is a grave matter, and can result in failing the course and/or more severe disciplinary action. See the section Academic Integrity in the Undergraduate Catalog[3] and

[3] `http://www.catalogs.umn.edu/ug/gen/policies.html\#integrity`

the Office for Student Academic Integrity (OSAI) web site[4].

In general, academic dishonesty (i.e. cheating) means misrepresenting the means by which your work was created — for example, claiming the work of another as your own. Academic dishonesty hurts you, because it diminishes your learning, and it hurts your classmates as well, because it diminishes the reputation of the department and the University — these reputations being the only things which lend value to awarded degrees.

The OSAI definition of academic dishonesty is as follows: "Scholastic dishonesty means plagiarizing; cheating on assignments or examinations; engaging in unauthorized collaboration on academic work; taking, acquiring, or using test materials without faculty permission; submitting false or incomplete records of academic achievement; acting alone or in cooperation with another to falsify records or to obtain dishonestly grades, honors, awards, or professional endorsement; altering, forging, or misusing a University academic record; or fabricating or falsifying data, research procedures, or data analysis."

In the context of this course, we will specify whether assignments are individual or group work. For the rest of this section, "you" refers to you as an individual or you as a group, as appropriate, and "others" refers to other individuals or groups. While you are free to discuss assignments with others, the work you turn in must be your own. This means that on written assignments you must create and write your own work, and on programming projects you must design, implement, debug, and test your program on your own. In addition, copying, assisting, or collaborating on an exam is misconduct, as is changing your answer after the exam is returned and then asking for regrading. Claiming emergency when none exists, in order to obtain special consideration on homework or exams, is also considered misconduct.

If you have any questions on this matter, please do not hesitate to ask us.

5.4 Class Conduct

Students are expected to treat their fellow students in the class, the instructor, and the teaching assistants with respect. This includes arriving at class on time and staying until the end of class (arriving late and leaving early are distracting to your instructor and classmates, and they interfere with group work). Talking to neighbors, reading newspapers, using a laptop for anything but course-related work, and sleeping during lectures are also distracting and disrespectful, so don't do these things.

5.5 Disabilities

It is University policy to provide, on a flexible and individualized basis, reasonable accommodations to students who have documented disability conditions (e.g., physical, learning, psychiatric, vision, hearing, or systemic) that may affect their ability to participate in course activities or to meet course requirements. Students with disabilities are encouraged to contact Disability Services and their instructors for a confidential discussion of their individual need for academic accommodations. Disability Services is located in Suite 180 McNamara Alumni Center, 200 Oak Street. Staff can be reached by calling 612-626-1333 voice or TTY, or on the web at `http://ds.umn.edu/`.

[4] `http://www.osai.umn.edu`