

CSC207 Algorithms and Object-Oriented Design

Spring 2009

Synopsis: This class introduces the object-oriented computational model, the third in our curriculum. We will learn about algorithms and representing data ownership and organization via objects and their collaboration.

MTWTF	11:00-11:50 pm	Science 3819
-------	----------------	--------------

Instructor: Jerod Weinman
Office: Noyce 3825
Phone: x9812
E-mail: [weinman]

Office hours:

Monday	3:00-4:00 PM
Tuesday	1:30-2:30 PM
Wednesday	4:00-5:00 PM
Friday	2:30-3:30 PM

or by appointment.

Course web page: <http://www.cs.grinnell.edu/~weinman/courses/CSC207/2009S/>

1 Overview

You have already studied the functional and machine-oriented imperative models of computation. In this course we will learn about the object-oriented model of computation. This abstraction facilitates a certain way of thinking that can help us solve many computational problems in a manner similar to the way we solve problems in our every day lives. It scales well and will facilitate our study of an important area of computer science: the organization, representation, and accessing of information. Thus, we will learn about some common ways of organizing data and the algorithms that allow efficient access to it. Such data structures are integral to many computational artifacts, from your computer's operating system, to genome sequencing, to GoogleTM-scale Internet-wide indexing.

Our major objectives for this course include:

- Fluency in object-oriented fundamentals
- Learning about object design patterns
- Sharpening algorithmic analysis skills
- Understanding how data organization affects access efficiency

Why take it? Understanding how to do efficient computation with the right data structures is a fundamental job of any computer scientist. In addition, object-oriented techniques have become a prevalent, widely used method of design and development. A thorough understanding of these tools will take you to the next level of maturity in computer science.

What do I need to know? This course assumes you have experience with a (moderately) typed imperative/procedural programming language (e.g., C, or even C++ or Java). Contrast this with Scheme, where a parameter to a procedure could be *anything*. In C, often the compiler will complain if you have mistaken a `double` type for an `int` type. You should also be familiar with some basic computational ideas like encapsulation and code re-use with procedures and recursion.

2 Textbook

Our course will be based on the following text:

Mark Allen Weiss. *Data Structures and Problem Solving Using Java*, Third Edition. Addison-Wesley, 2006. ISBN: 0-321-32213-4

The online Java API reference is generally sufficient, but for those who prefer hefty paper references, and some commentary to boot, the following can be a useful guide. (Student ACM members have online access through Safari.)

David Flanagan. *Java in a Nutshell*, Fifth Edition. O'Reilly, 2005. ISBN 0-596-00773-6.

3 Class Attendance

Class meetings will involve a mix of discussions, lectures, and lab activities. In short: *You are expected to attend and actively participate in class. I am expected to make class worth attending.*

I know that sometimes “things happen.” Therefore, you will be granted **one** unexcused absence from class without penalty. However, this is a collaborative discussion and lab-based course, so your presence is integral to your learning. Thus, the following will be deducted from your participation grade in the case of absences:

2-3 absences	20%
4-5 absences	40%
6 or more absences	60%

If you wish me to acknowledge your absence as excused, you must either:

1. Notify me at least 7 days in advance to make arrangements for your absence, *or*
2. Ensure I have received documentation of your absence’s circumstances *post hoc* from Health Services or Student Affairs.

It is very important for you to attend class on lab days because you will often work collaboratively. In addition, our discussions in non-lab days benefit from your contributions. If you do miss a class, you must first talk to a classmate about any material that you may have missed. After that, you may follow up with me about any further questions or concerns.

4 Schedule of Topics

The following is an *approximate* schedule of topics to be discussed during the course. See the web page schedule for daily details.

Week	Topic
1	Introduction to Java and OOP
2	Objects and Classes
3	Inheritance and Generics
4	Design Patterns
5	Algorithm Analysis
6	Using the Collections API
7	Recursion

Week	Topic
8	Sorting
9	Implementing Collections
10	Linear Structures
11	Trees
12	Binary Search Trees
13	Hash Tables
14	Priority Queues

5 Assignments and Activities

Under a normal 16 credit load, I expect that you will spend at least 40 hours per week on your studies (class time, homework, and studying). Thus, you should plan to spend a minimum of 10 hours/week on work for this course (but hopefully more). With class time clocking in at $3\frac{1}{3}$ hours, you’ll have $6\frac{2}{3}$ hours/week left for the following:

5.1 Reading

You should check the class schedule for updates and read any material that has been assigned *before* coming to class. Reading the textbook entails the following:

Overview You should skim through the reading once to get an overview of the material to be covered, paying particular attention to subject headings and topic introductions. This first “reading” can (and should) be very quick. (*Expected time: 5 to 10 minutes.*)

In-Depth Next, read the material closely. Try to understand what individual steps of algorithms or mathematical proofs are accomplishing. Not everything will make sense at this point, but hopefully many things will. (*Expected time: 40 to 50 minutes.*)

Final Notes After carefully reading the material, mentally review and try making a few notes to yourself about what you think are the most important concepts being covered, as well as any questions you have. (*Expected time: 5 to 10 minutes.*)

Many of the readings are fairly short (about 13 ± 6 pages, or roughly 30 pages per week), but can contain mathematics or source code that require a moderate amount of study. While I realize not everyone learns best by reading, you are asked to make your best effort and come to class with any questions you may have. Then we can proceed with examples that enhance and clarify the material in class.

5.2 Laboratory Activities

Many class days will involve collaborative laboratory work to experiment with and learn more about the concepts we discuss. You may not always complete the laboratory assignments during class. It is important that you finish these outside of class to be sure that you are engaging in all the material we cover. Like playing an instrument or speaking a foreign language, the only way to become proficient is to practice, practice, practice!

5.3 Homework Assignments

Over the course of the term, there will be approximately nine programming assignments addressing problems from lecture material and laboratory exercises. These will be due weekly in the first half of the term and less frequently in the second half. Instructions regarding collaboration will be given with each assignment. Unless the assignment specifies that group work is allowed, you are welcome to discuss material with others, but any work you do and submit should be your own. (One good rule of thumb is that you should not leave a discussion with written material regarding the assignment.)

Discussion of concepts and approaches with other classmates is encouraged. Debugging programs can be difficult and is often helped by explaining your code to someone else (which will also frequently help you to explain the bug to *yourself*). All such contributions by others (not in your group) should be properly attributed in your report. Furthermore, all the work you submit (code, experimental data, write-ups, etc.) must be that of the group. Code provided by the instructor or the text should be attributed, but no other code or written work (from any source) may be shared with others or copied for your own use.

5.4 Exams

As opportunities for you to demonstrate your object-oriented design prowess and grasp of data organization principles there will be two hour exams as well as a cumulative final exam. In addition, we will have a brief (half-period) Java quiz at the end of week three to make sure everyone is up to speed on some important, basic principles of the programming language.

Java Quiz	Friday, February 26
Hour Exam 1	Friday, March 6
Hour Exam 2	Friday, April 17
Final Exam	Wednesday, May 13 (2 PM)

Do not make airline reservations that will conflict with your final exam schedule.

You are highly encouraged to use the PioneerWeb Discussion Board for questions related to the course or laboratory exercises. If a post is related to an assignment, it must adhere to the standards of collaboration for that particular assignment. If you are debugging, one good rule of thumb is to not post code; however, it is possible that error messages may be enlightening to others. (I often find that when I am explaining some error by writing up a post or email, I typically find its source.)

6 Grading

My goal is for everyone taking this course to be able to demonstrate familiarity and fluency with the course concepts. I would be very happy if you all met the goals above and received “A”s. The following weighting will provide a basis for evaluation.

Homework Assignments	40%
Participation	10%
Java Quiz	5%
Exams	30%
Final Exam	15%

7 Academic Honesty

You, as students, are members of the academic community. Both the College and I expect the highest standards of academic honesty. (See the Grinnell College Student Handbook, e.g.,

<http://www.grinnell.edu/offices/studentaffairs/shb/section3/academichonesty>).

Among other things, this means clearly distinguishing between work that is your own, and work that should be attributed to others. It is expected that the collaboration policies given in this syllabus and on particular assignments will be followed. Furthermore, any program results or output must be faithfully recorded, not forged. (A thoughtful explanation of unexpected behavior can often be a worthwhile submission and is *much* better than the alternative.)

In your homework assignments, you must give specific attribution for any assistance you receive. For example, one possible acknowledgment format is “[*Person X*] helped me to do [*thing Y*] by explaining [*Z*].”

As an instructor, I will meet my obligation to bring any work suspected to be in violation of the College’s Academic Honesty Policy to the attention of the Committee on Academic Standing, after which there is no recourse with me.

8 Deadlines

Assignments are due at the *beginning* of class on the specified date. Work submitted more than ten minutes after the beginning of class will be considered late. Assignments due on days for which you have a prior excused absence must still be submitted by the deadline.

A late penalty of 33.33% will be deducted at each subsequent class meeting. Thus, you have at most two additional meetings to submit your work for a non-negative grade.

Exception: Deadlines for MathLAN computer-based assignments will automatically be extended by at least one class period if MathLAN is down for an unscheduled period of 3 or more hours during the week preceding the assignment due date.

9 Contacting Me

Please come by during my office hours to discuss the course content, get any extra assistance, or just talk about how the course is going. Note that if multiple students have similar questions or issues, we may work together as a group. If you cannot attend a scheduled office hour, you may also email me to schedule an appointment; please include 3-4 possible meeting times so that I can pick one that works for me.

I enjoy getting to know my students, but I prefer to reserve office hours for academic matters. If you would like to have a more informal conversation, I would be delighted to accept an invitation to lunch (<http://www.grinnell.edu/offices/studentaffairs/chaplain/religiousandspiritualprograms>).

Email is also a reliable way to contact me, but please allow 24 hours for a response (except on weekends, when I do not regularly read email). You may also call me in my office (x9812).

10 Accommodations

If you have any disability that requires accommodations, please meet with me right away so that we can work together to find accommodations that meet your learning needs. You will also need to provide documentation of your disability to the Dean for Student Academic Support and Advising, Joyce Stern, located on the 3rd floor of the Rosenfield Center (x3702).

With thanks to Janet Davis for the “Reading Suggestions” and other key policies.