In this class we will cover the essential data structures that every Computer Scientist needs to have at their fingertips. We will also work on some of the introductory material for your latter classes. This class forms the foundation for all your future studies in Computer Science, and your progress in this class will likely be reflected in your later classes. Study and work hard here and it will pay dividends in the future. We will study representations and operations on basic data structures including arrays, linked lists, stacks, queues, and recursion; binary search trees and balanced trees; hash tables, dynamic storage management; introduction to graphs. All course work and examples are taught using Java and the Eclipse IDE.

**Enrollment Information**

Prerequisites: Computer Science 108 and Mathematics 245.

You should be comfortable programming in an Object-Oriented environment as you will be expected to handle multiple classes or methods.

**Course Materials**


- Course Reader, by Alan Riggins. Required.


**Course Structure and Conduct**

The course will be taught as a blended course including lectures, discussions, and online classes. The class will meet most weeks, and there will be a discussion of the topics of the week. Some weeks, this will be a traditional lecture, while some weeks the class will meet to discuss an online video lecture. The class will utilize clickers to assess student progress.
Computing environment

For this course, you will do extensive programming in Java, the language of instruction for the course. We will be using Java SE 1.6 or 1.7. Each student will be given a class account on rohan, the University academic Sun computer for submitting the work. You may wish to develop your programs on your home computer, but all programming assignments must compile and run in your class account on rohan. Please note: the class accounts are deleted at the end of the semester; you are also eligible for a personal account on rohan, but you can not use that for turning in your assignment.

All work will be done in eclipse, and work will be submitted using CVS, the concurrent versioning system. Oracle released version 1.7 of the JDK, which introduces Java SE 7. This has only a few updates and minor changes from Java 1.6. Since, for grading purposes, all programs must compile on rohan we will use Java SE 1.6 // JDK 1.6.

The current version of Java on rohan (as of Jan, 2014) is:
$ java -version
java version "1.6.0_65"
Java(TM) SE Runtime Environment (build 1.6.0_65-b14)
Java HotSpot(TM) Client VM (build 20.65-b04, mixed mode, sharing)

Course Assessment and Grading

There will be three programming assignment worth 11% of the total each. The first assignment (assignment 0) will be worth 2% of the final grade.

There will be two midterm exams and a final. The exams will be weighted so that the two midterms are worth 15% of the final grade each, and the cumulative final is worth 30% of the final grade.

Your grade will be based on programs (35%), exams (60%), and participation (5%)

The minimum grading scale will be:
- 90% and above: A-, A
- 80%-89%: B-, B, B+
- 70%-79%: C-, C, C+
- 60%-69%: D-, D, D+
- below 60%: F

Depending on overall progress of the class, this scale may be adjusted (curved). This will be discussed in class.

Course Policies and Advice

The class blackboard page will be the main source of information, and assignments for the class. Make a point of visiting the web page before each class session.

Each student will receive a class account on rohan for use in this class. You should use this account only for work related to this class. As this is a class account, your instructors or the TAs may enter your account at any
time, and nothing in your account should be considered private. It is against University policy to use SDSU computer resources for commercial purposes. All SDSU students may apply for a personal account on rohan. You should use a personal account for all work not related to this course.

By default, the permissions set on your account files are private. Do not set global permissions on any files in your class account. This is a gradeable issue.

You should safeguard your password, and protect access to your class account. Be vigilant when working in the computer labs. Lock the screen, or log out if you must step away from the lab. You are responsible for protecting your intellectual property.

You may change the password on your class account if you wish. However, do not modify your .login, .bashrc, .ssh or other configuration files unless you know what you're doing. If you accidentally change something that makes it impossible for you to gain access to your account, it may take several days to fix. You will not be given an extension if you caused the problem.

Some folders and other settings files provide the instructors and TAs with access to your account without using your password. Under no circumstances should you modify or delete this folder and file. I will log in to your class accounts to retrieve program files for grading. If you have done something to your settings files or folders and I cannot retrieve your program files for an assignment, you will not receive any credit for that assignment and it will be up to you to fix the account before the next assignment is due.

I have developed an automated system for collecting and running the programs that are assigned. It is very important that you follow the assignment directions precisely. In particular, make sure that you use the exact filename, and the exact subdirectory specified in the assignment. Remember that UNIX filenames are case-sensitive. If the grading program cannot find your program files, you will lose credit.

Programming assignments may be accepted up to 7 days late, but with a 5% per day penalty. Plan ahead, and start early. If you wait until the last possible day, the maximum grade you will get is 65% (a D). There will be no makeup exams. If you must miss a midterm exam, you will need to get permission from me first, or provide a satisfactory (my discretion) reason for having missed an exam.

I do not debug programs via email. Do not send me an email asking me to tell you what's wrong with your program (that's what office hours are for).

Please do not send me emails asking questions for which answers are easily available.

Email is not a reliable form of communication. I receive many hundreds of emails a week, and depending on when yours arrives I may or may not address it. Just because you send me an email does not mean I have to answer it! I will be in class, and both I and the TAs will have office hours to discuss problems.

**Other Course Policies**

**Cheating**

You must design and write your own programs. Exchanging ideas is fine. However, you must not exchange code. During the grading process, I will examine your code carefully. Anyone caught cheating, either on a programming assignment or on a test, will receive an "F" and will be turned in to the Center for Student Rights and Responsibilities.
You are responsible for protecting your intellectual property. If someone acquires a copy of your program due to carelessness on your part, you will be held accountable.

**Plagiarism**

Plagiarism is using someone else's work without proper attribution. It doesn't matter if that work is online or your friends. You may also not use work you submitted to other universities or other classes at San Diego State University without the express consent of the instructor. The instructor will not tolerate any plagiarism. If I catch you cheating you will fail the entire course. There will be no make up and no excuses accepted. Your cheating will also be reported to the Center for Student Rights and Responsibilities. Remember, they are there for you as much as for me.

**TurnItIn or other plagiarism detection software**

Students agree that by taking this course all required papers may be subject to submission for textual similarity review to Turnitin.com or another similar website or software database for the detection of plagiarism. All submitted papers will be included as source documents in the Turnitin.com reference database (or another similar database) solely for the purpose of detecting plagiarism of such papers.

You may submit your papers in such a way that no identifying information about you is included.

Another option is that you may request, in writing, that your papers not be submitted to Turnitin.com. However, if you choose this option you will be required to provide documentation to substantiate that the papers are your original work and do not include any plagiarized material.

If you do not agree to this, please see Dr. Edwards as soon as possible.

**MOSS**

Students agree that by taking this course all required code may be subject to submission for similarity review to MOSS or another similar automated code analysis program. Code may be included as source documents in the MOSS reference database (or another similar database) solely for the purpose of detecting plagiarism of such code. You may submit your code in such a way that no identifying information about you is included.

Another option is that you may request, in writing, that your code not be submitted to MOSS or similar sites. However, if you choose this option you will be required to provide documentation to substantiate that the code is your original work and do not include any plagiarized material.

If you do not agree to this, please see Dr. Edwards as soon as possible.