Subject: CSCI    Number: 262
Course Title: Data Structures
Sections: A & B
Semester/year: Spring 2016

Section A:
  Instructor: Christopher Painter-Wakefield
  Contact information:
    office: Brown Building (BB) 280I
    phone: 303-273-3717
    email: cpainter@mines.edu
  Office hours: TTh 1-3pm, by appointment, or whenever my office door is open
  Class meeting days/times: MWF 1:00-1:50
  Class meeting location: MZ 026

Section B:
  Instructor: Daniel Hathorn
  Contact information:
    office: TBA
    phone: TBA
    email: TBA
  Office hours: TBA
  Class meeting days/times: MWF 4:00-4:50
  Class meeting location: MZ 026

Both Sections:
  Web Page/Blackboard link:
    Course Website: http://eeecs.mines.edu/Courses/csci262/
    Blackboard: https://blackboard.mines.edu/
    Piazza: https://piazza.com/mines/spring2016/csci262
Instructional activity:  3  hours lecture  0  hours lab  3  semester hours

Course designation:  ___ Common Core  ___ Distributed Science or Engineering

  X  Major requirement  ___ Elective  ___ Other (please describe ___________)

Course description from Bulletin:

Defining and using data structures such as linked lists, stacks, queues, binary trees, binary heap, hash tables. Introduction to algorithm analysis, with emphasis on sorting and search routines. Language skills: abstract data types, templates and inheritance. Prerequisite: CSCI261 with a grade of C- or higher. 3 hours lecture; 3 semester hours.

Textbook and/or other requirement materials:

Required text: Programming Abstractions in C++ by Eric S. Roberts

Student learning outcomes: At the conclusion of the class students will be able to:

1. Evaluate and understand the trade-offs in selecting one data structure vs. another, and employ data structures in the solution of realistic problems.
2. Understand the purpose of an abstract data type (ADT) such as a list or map. Recognize that ADTs can have different implementations.
3. Implement fundamental data structures including linked structures, stacks, queues, hash tables, and trees.
4. Implement recursive subroutines to search and manipulate data structures.
5. Determine computational complexity of simple functions, including asymptotic analysis of upper and average complexity bounds, big O notation, standard complexity classes, and empirical measurements of performance.
6. Understand the behavior and computational characteristics of fundamental computing algorithms including sorting algorithms, hashing, graph representation and traversal, and binary search tree operations.
7. Implement programs using object-oriented design and object-oriented programming including encapsulation and information hiding, separation of specification and implementation; classes, subclasses, and inheritance; polymorphism; class hierarchies.

Brief list of topics covered:

1. Abstract data types
2. Data structures, including linked structures, stacks, queues, hash tables and trees
3. Pointers and dynamic memory
4. Classes, inheritance, and polymorphism
5. Standard template library (STL)
6. Recursion
7. Searching
8. Sorting

Policy on academic integrity/misconduct: The Colorado School of Mines affirms the principle that all individuals associated with the Mines academic community have a responsibility for establishing, maintaining an fostering an understanding and appreciation for academic integrity. In broad terms, this implies protecting the environment of mutual trust within which scholarly exchange occurs, supporting the ability of the faculty to fairly and effectively evaluate every student’s academic achievements, and giving credence to the university’s educational mission, its scholarly objectives and the substance of the degrees it awards. The protection of academic integrity requires there to be clear and consistent standards, as well as confrontation and sanctions when individuals violate those standards. The Colorado School of Mines desires an environment free of any and all forms of academic misconduct and expects students to act with integrity at all times.
Academic misconduct is the intentional act of fraud, in which an individual seeks to claim credit for the work and efforts of another without authorization, or uses unauthorized materials or fabricated information in any academic exercise. Student Academic Misconduct arises when a student violates the principle of academic integrity. Such behavior erodes mutual trust, distorts the fair evaluation of academic achievements, violates the ethical code of behavior upon which education and scholarship rest, and undermines the credibility of the university. Because of the serious institutional and individual ramifications, student misconduct arising from violations of academic integrity is not tolerated at Mines. If a student is found to have engaged in such misconduct sanctions such as change of a grade, loss of institutional privileges, or academic suspension or dismissal may be imposed.

The complete policy is online.

**Collaboration Policy for Programming Projects in EECS Courses:**
The following policy exists for all CS courses in the EECS department. This policy is a minimum standard; your instructor may decide to augment this policy.

1. If the project is an individual effort project, you are not allowed to give code you have developed to another student or use code provided by another student. If the project is a group project, you are only allowed to share code with your group members.

2. You are encouraged to discuss programming projects with other students in the class, as long as the following rules are followed:
   a. You view another student's code only for the purpose of offering/receiving debugging assistance. Students can only give advice on what problems to look for; they cannot debug your code for you. All changes to your code must be made by you.
   b. Your discussion is subject to the empty hands policy, which means you leave the discussion without any record [electronic, mechanical or otherwise] of the discussion.

3. Any material from any outside source such as books, projects, and in particular, from the Web, should be properly referenced and should only be used if specifically allowed for the assignment.

4. If you are aware of students violating this policy, you are encouraged to inform the professor of the course. Violating this policy will be treated as an academic misconduct for all students involved. See the Student Handbook for details on academic dishonesty.

**Instructor's Addendum:**
As an addendum to this policy, you are required to submit a README file with every (non-lab) assignment. The README should list every person (other than the instructor or a TA) who assisted you in some way on the assignment. The README is also the place to give appropriate credit to any outside source that contributed to your submission.

**Grading Procedures:** Programming assignments come in two flavors: labs, which consist of small, automatically graded programs, and regular assignments, which will be fully functional programs to accomplish some specified task. There will be two midterm exams and a final. A small percentage of your grade will be based on your participation in classroom activities and on Piazza.

Grading will be done on an absolute, but adjustable scale. This means that there is no curve. Anyone earning 90% or more of the total number of points available will receive a grade of A; 80% or higher a B, etc. This scale may go down, but it will not go up.

**You must pass the final (60% or higher) to pass this course.**

All grades will be posted in Blackboard.

- Assignments: 40%
- Labs: 10%
- Class participation: 5%
- Midterms: 20%
- Final exam: 25%

**Coursework Return Policy:** Homeworks and exams will be graded and returned as quickly as possible, generally within two weeks.
**Absence Policy:** Please see the Mines policy on absences at [http://inside.mines.edu/Student-Absences](http://inside.mines.edu/Student-Absences)

**Homework:** Late work is *strongly* discouraged. All work will be accepted within 4 working days after the deadline (weekend and holiday days do not count in the 4 days). Students will lose 10 percentage points per day late. After 4 days, the work will not be accepted at all.