COURSE OUTLINE

CS 2420
Introduction to Data Structures and Algorithms
Spring Semester 2015
Weber State University

ADMINISTRATION

Instructor: Dr. Brian Rague
Email: brague@weber.edu
Web Page: http://faculty.weber.edu/brague
Phone: 626-7377
Office Location: TE110-F
Office Hours: MW from 9:30am-11:30am, T from 10:30am-11:30am

Time and Room: 7:30 – 9:20 MW in TE103C (with Lab)

Text: Data Structures Using C++ (2nd ed.) by D. S. Malik (Ma)
Pre-requisites: CS1410
Pre-req/Co-req: MA1080 or MA1050/MA1060

COURSE DESCRIPTION

General principles of common data structures and design of efficient algorithms. Topics include: arrays, linked-lists, stacks, queues, trees, graphs, tables, storage and retrieval structures, searching, sorting, hashing, and algorithmic analysis. Emphasis will be on abstraction, efficiency, re-usable code, and object-oriented implementation.

LEARNING OBJECTIVES

- To learn adaptable skills in a software development workplace environment.
- To learn and understand object-oriented program design.
- To learn and understand algorithm development techniques.
- To learn and understand the foundational conceptual design as well as the implementation of data structures including static and dynamic arrays, vectors, linked lists, stacks, queues, and binary trees.
- To become proficient at comparing and contrasting efficiency both in time and space for these data structures.
- To learn and understand common search and sorting algorithms.

CLASS PROCEDURE

Presentations of course concepts and fundamentals are recorded as mp4 videos. Students are required to view all recordings assigned for a given week and complete a short quiz associated with each recording. The weekly recordings and quizzes are accessible.
through the Canvas site for this course. Grading for presentations is credit/no-credit; credit for viewing each recorded presentation is earned only when all questions in the corresponding quiz are answered correctly for a single submission of the quiz. You can take the quiz as many times as you wish in order to receive credit, but it must be completed no later than Tuesday midnight following the week in which the recordings are assigned.

During all Wednesdays of the course semester (with the exception of Exam Weeks 5 and 10), an in-class activity will take place that requires student attendance in order to receive credit. In-class activities involve either (1) an individual and team assessment of course topics with sufficient review prior to these assessments or (2) a small-team programming exercise to be completed and demonstrated to the instructor prior to the conclusion of the class session. See the course schedule below for specific activity descriptions.

During all Mondays of the course semester, informal discussions and assistance regarding individual programming assignments and course topics will take place on an as-needed basis, depending on individual student requests. Students are not required to attend these sessions, which are designed specifically to assist students experiencing challenges with course concepts or programming assignment tasks.

GENERAL EXPECTATIONS

Attendance

CS 2420 covers a broad range of information that requires some form of interaction and participation in the classroom setting. As noted above, a good portion of the final grade includes in-class assessments and small-team program design and development. It is vital that you attend all required classes. Your punctuality and attentiveness (not reading newspapers, not surfing the web) is appreciated and will be noted. Activated cell phones, because of their capacity to generate noises and to be spoken into, are not allowed. Students are responsible for material covered during absences regardless of the reason for the absence.

Academic Honesty

Academic honesty is highly valued at Weber State University and within this class. A student must always submit work that represents his or her original words or ideas. If any words or ideas are used that do not represent the student's original words or ideas, the student must cite all relevant sources. The student should also make clear the extent to which such sources were used. Words or ideas that require citations include, but are not limited to, all hardcopy or electronic publications, whether copyrighted or not, and all verbal or visual communication when the content of such communication clearly originates from an identifiable source.

Individuals involved in any acts of cheating or plagiarism will be given a failing grade for the course. In addition, names of these individuals will be submitted for disciplinary action by the department and the university.

Academic dishonesty involves any and all of the following:

- Having a tutor or friend complete a portion of your assignments
- Having a reviewer make extensive revisions to an assignment
• Copying work directly from another student
• Using information from online information services without proper citation

Team programming is not discouraged but any collaboration should be at the conceptual level. If detail source code is duplicated from one program into another, it will be deemed as cheating and severe action will be taken against persons who knowingly have participated. Source code that doesn’t match the output results may also be deemed as cheating which may result in severe penalties.

Students are expected to be familiar with the WSU Student Code and abide by it. The Code may be reviewed on line at http://www.weber.edu/ppm/6-22.htm (pay specific attention to Section 4D). All necessary steps will be taken to enforce the Student Code to guarantee fairness to all students.

CS Department policy dictates that any verifiable evidence of student academic cheating, as defined and determined by the instructor, will result in: 1) an automatic failing grade for the class and 2) a report to the Dean of Students that will include the student’s name and a description of the student’s dishonest conduct.

Readings, Video Recordings, Assignments, & Labs

Because in-class assessments are based primarily on material in the course text and the recorded video presentations, it is highly recommended that you complete the assigned weekly readings and recorded presentations prior to taking the assessment. There are 6 individual programming assignments. The individual programs will be graded on proper documentation, successful compilation, execution, and satisfaction of requirement specifications as given in the assignment description. Due dates for individual assignments are provided in the class schedule below. Each student will upload assignments to a designated class server, typically using the current WSU Online portal. The uploading procedure will be explained during the first week of the term. Assignments may not be submitted by email.

The computing labs, in-class computers, and tutors are available to help you complete your assignment. I will be present during most labs scheduled during class time to assist you. Use this lab time primarily to collaborate with fellow team members and to complete the group projects.

C++ Compilers

As part of this course, you will need a C++ compiler. Your programs will need to be compiled to run on a Microsoft Windows Operating System. Try to introduce yourself to a C++ programming package as early as possible.

Microsoft options for compilers include Visual Studio 2010 and Visual Studio 2012. I will test your programs using Microsoft Visual Studio 2010/2012, or Linux g++.

The WSU CS department has an Academic Alliance program with Microsoft that allows students to obtain Microsoft Visual Studio for free. The student chapter of ACM handles the ordering and distribution of this software.

The website for the MSDNAA is http://msdnaa.cs.weber.edu/. If DNS issues prevent the site
from being visible outside of the main campus labs, alternatively you can try http://137.190.19.14 .

You can also find ANSI standard C++ compilers on the web for free. Any ANSI compliant compiler should work. 'Bloodshed' is an aptly named example. If you find one that works well, please post so that others may benefit.

In order for your programs to run on my compiler, most of the libraries should NOT have the '.h' on the end and the 'namespace' statement must be present. For example:

```
#include <iostream> // for standard i/o
#include <iomanip> // for formatting
#include <cmath> // for sqrt, pow, etc
using namespace std;
int main ( )
```

**Important Information about Your Programs**

Programming assignments should be uploaded using the WSU Online site assigned to this class. Details will be given during class. Typically only the source code (.cpp file) should be submitted. Executables are rarely required. I will compile the submitted source in order to confirm that no syntax errors exist within the code. **All submitted programs should compile as a minimum requirement.**

Avoid submitting programs by email, because there’s always a distinct possibility I may not receive the assignment due to some obscure networking, server, or mail application problem. I will not confirm receipt of assignments emailed to me.

Collaboration on individual assignments should be at the conceptual level only. If detail source code is duplicated from one program into another, it will be deemed as cheating and severe action will be taken against persons who knowingly have participated. Students should ensure that the latest version of source code is submitted. Source code that doesn’t match the output results may also be deemed as cheating which may result in severe penalties.

**Documenting Your Programs**

Program documentation is very important. For CS2420, we will emphasize explicit documentation for individual code, as described in the section below.

**I. Individual code**

Program documentation for individual assignments will consist of both prologue and explanatory comments. Please note the following three requirements regarding the documentation for your programs.

**1. The comment headers for each of your submitted programs must adhere to the following format:**

```
// John Doe
// CS 2420 – 7:30 am
```
Assignment #99
Dr. Rague
Due: mm/dd/yy
Version: 1.0

This program calculates the revenue generated by an employee installing coaxial cable

---

// This program calculates the revenue generated by an employee installing coaxial cable

// Compiler directives
#include <iostream>

using namespace std;

---

2. Each main modular unit (classes and functions) should be preceded by a short description as shown here:

---

// This class represents a circle that can be displayed on a console window.

---

class Circle
{
  protected:
    int x, y;

---

3. You should also include appropriate in-line comments throughout your code to clarify any sections or statements in which the programming strategy isn't immediately obvious.

---

// Variable declarations
int Num_Cases;             // The number of cases of data values to process
int Installations;        // The number of installations
double Yards_Of_Cable,    // The yards of cable installed
                     Feet_Of_Cable, // The number of yards of cable converted to feet
                     Revenue;       // The revenue generated for this case of data

---

Late Assignment Policy

Please note that a late assignment will be accepted only within the first five days following its original due date. The maximum grade possible for a late assignment will be automatically reduced by an amount equal to 10% of the original total point value for each day it is late, and, as stipulated, will not be accepted at all after five days. No assignments will be accepted after the last day of class, prior to finals week. No exceptions.

“I really, really need to get a C” Policy:

The most effective method for obtaining a C or above in this class is to submit assignments when they are due and to stay current with course topics. The curriculum is carefully designed to fit the number of course weeks. In order to uphold academic rigor and integrity, student
grades must be based on the degree to which the course requirements listed in the syllabus are fulfilled. Except when specifically made available per the course schedule, extra credit assignments are not allowed. If you approach me anytime during the term claiming that special allowance should be made because you need a C to move forward in the program, graduate, receive financial aid, etc., I will decline your request and refer you to this clearly worded policy.

Exams & Quizzes

There will be six individual/group in-class assessments, and three exams. The in-class assessments will be composed of multiple choice, short answer, and true/false questions. Each student will take the test individually, and then each team will take the same test as a group. These assessments are primarily used to test your knowledge of key terminology and programming concepts that have been provided in the reading and highlighted in the class presentations. To receive credit, each student must take the assessment both individually and as a team member. As long as you read, watch the presentation videos, and attend the required class sessions, you should have no problem receiving a good score on the tests.

Assessment 1 – Week 2
Assessment 2 – Week 4
Assessment 3 – Week 7
Assessment 4 – Week 9
Assessment 5 – Week 12
Assessment 6 – Week 14

Exam 1 – Week 5
Exam 2 – Week 10
Exam 3 – During Finals Week

The exams will be given at a Chi-Testing site on an affiliated campus. Information about how the exams will be administered will be given during the week before the exam. There will be no weekly assignments during the week of an exam. The exams will consist of multiple choice, True/False, short answer questions, and programming exercises.

The exams are not to be shared or discussed with other students. If I find out or discover that there has been any cheating, it will result in an automatic failure for the course!

Each student will receive the grade, which in the best judgment of the instructor, he or she has earned. Requests for a grade higher than the one earned, for any reason (including loss of student visa, entrance requirements into a professional school, etc.) will not be honored. There will be no retests. Make-up exams will be given only for valid reasons and if the instructor is notified in advance of the scheduled exam hour.

Grading Policy

<p>| Individual Assignments | 30% |
| Presentations Videos/Quizzes | 15% |
| Individual/Group Assessments | 10% |
| In-class Exercises | 15% |
| Exams | 30% |</p>
<table>
<thead>
<tr>
<th>LETTER GRADE</th>
<th>PERCENT RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>total &gt;= 93.3%</td>
</tr>
<tr>
<td>A-</td>
<td>90.0% &lt;= total &lt; 93.3%</td>
</tr>
<tr>
<td>B+</td>
<td>86.7% &lt;= total &lt; 90.0%</td>
</tr>
<tr>
<td>B</td>
<td>83.3% &lt;= total &lt; 86.7%</td>
</tr>
<tr>
<td>B-</td>
<td>80.0% &lt;= total &lt; 83.3%</td>
</tr>
<tr>
<td>C+</td>
<td>76.7% &lt;= total &lt; 80.0%</td>
</tr>
<tr>
<td>C</td>
<td>73.3% &lt;= total &lt; 76.7%</td>
</tr>
<tr>
<td>C-</td>
<td>70.0% &lt;= total &lt; 73.3%</td>
</tr>
<tr>
<td>D+</td>
<td>66.7% &lt;= total &lt; 70.0%</td>
</tr>
<tr>
<td>D</td>
<td>63.3% &lt;= total &lt; 66.7%</td>
</tr>
<tr>
<td>D-</td>
<td>60.0% &lt;= total &lt; 63.3%</td>
</tr>
<tr>
<td>F</td>
<td>total &lt; 60.0%</td>
</tr>
</tbody>
</table>

Students with Disabilities

Any student requiring accommodations or services due to a disability must contact Services for Students with Disabilities (SSD) in room 181 of the Student Services Center. SSD can also arrange to provide course materials (including this syllabus) in alternative formats if necessary.

WSU Student Code

Students are expected to be familiar with the WSU Student Code and abide by it. The Code may be reviewed online at [http://www.weber.edu/ppm/6-22.htm](http://www.weber.edu/ppm/6-22.htm) (pay specific attention to Section 4D). All necessary steps will be taken to enforce the Student Code to guarantee fairness to all students.

Bad Weather Policy

Use good judgement in the event of bad weather. Do not take unnecessary risks. The attendance and participation policies will be relaxed in the event of inclement weather. Exams will be rescheduled as necessary. Keep informed about class cancellations by listening to the radio.

Disclaimer

The following syllabus is subject to change at any time. The listing represents a tentative class schedule for the term and the topics covered. Alterations made in class supersede this document.
# CS 2420 Syllabus:

<table>
<thead>
<tr>
<th>WEEK</th>
<th>STARTING</th>
<th>TOPICS</th>
<th>READING</th>
<th>ASSIGNMENTS/ACTIVITIES</th>
</tr>
</thead>
</table>
| 1    | Monday, Jan 12 | Introduction, C++ basics review, classes                              | Chapter 1     | **Mon:** Groups Formed, Practice Assessment  
**Wed:** In-class Activity #1                                                     |
| 2    | Monday, Jan 19 | ADT’s, inheritance overloading, friends, templates                     | Chapter 2     | **Wed:** Assessment #1 on Week 1 & 2 Readings/Presentations  
*Assignments:* Indiv. Assignment #1–Due Monday, Week 3                           |
| 3    | Monday, Jan 26 | ADT’s, inheritance overloading, friends, templates                     | Chapter 2     | **Wed:** In-class Activity #2                                                           |
| 4    | Monday, Feb 2  | Pointers, heap, static and dynamic arrays                              | Chapter 3     | **Wed:** Assessment #2 on Week 1-4 Readings/Presentations  
*Assignments:* Indiv. Assignment #2–Due Monday, Week 5                           |
| 5    | Monday, Feb 9  | STL: containers, adapters, iterators, vector, list.                    | Chapter 4     | **Exam 1 on Chapters 1-4**                                                               |
| 6    | Monday, Feb 16 | Recursion                                                              | Chapter 6     | **Wed:** In-class Activity #3                                                           |
| 7    | Monday, Feb 23 | Linked lists, doubly-linked lists                                      | Chapter 5     | **Wed:** Assessment #3 on Week 5-7 Readings/Presentations  
*Assignments:* Indiv. Assignment #3–Due Monday, Week 8                           |
| 8    | Monday, Mar 2  | Stacks: linear and linked implementations                              | Chapter 7     | **Wed:** In-class Activity #4                                                            |
| 9    | Monday, Mar 16 | Queues: linear and linked implementations                              | Chapter 8     | **Wed:** Assessment #4 on Week 8-9 Readings/Presentations  
*Assignments:* Indiv. Assignment #4–Due Monday, Week 10                           |

**Spring Break: Week of Mar. 9**
<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Topic</th>
<th>Chapter</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Monday, Mar 23</td>
<td>Search Algorithms</td>
<td>Chapter 9</td>
<td><strong>Exam 2 on Chapters 5-9</strong></td>
</tr>
<tr>
<td>11</td>
<td>Monday, Mar 30</td>
<td>Hash Algorithms</td>
<td>Chapter 9</td>
<td><strong>Wed: In-class Activity #5</strong></td>
</tr>
<tr>
<td>12</td>
<td>Monday, Apr 6</td>
<td>Sort Algorithms</td>
<td>Chapter 10</td>
<td><strong>Wed: Assessment #5 on Week 10-12 Readings/Presentations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Assignments:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Indiv. Assignment #5– Due Monday, Week 13</td>
</tr>
<tr>
<td>13</td>
<td>Mon, Apr 13</td>
<td>Tree search and balancing, Binary Trees, AVL Trees, B-Trees</td>
<td>Chapter 11</td>
<td><strong>Wed: In-class Activity #6</strong></td>
</tr>
<tr>
<td>14</td>
<td>Mon, Apr 20</td>
<td>Graphs</td>
<td>Chapter 12</td>
<td><strong>Wed: Assessment #6 on Week 13-14 Readings/Presentations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Assignments:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Indiv. Assignment #6– Due Monday, Finals Week</td>
</tr>
</tbody>
</table>

**Finals Week**

**Week of Apr. 27**

**Exam 3 on Chapters 9-12**