# CS 1410 Spring 2013 Syllabus
## Object-Oriented Programming Using C++

Your goal must be to learn, my goal must be to help you learn

<table>
<thead>
<tr>
<th><strong>Instructor:</strong></th>
<th>Delroy A. Brinkerhoff, Ph.D.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Office Hours:</strong></td>
<td>1:30 - 2:30 MW and 11:30 - 1:30 T</td>
</tr>
<tr>
<td><strong>Office Location:</strong></td>
<td>TE 111A</td>
</tr>
<tr>
<td><strong>Phone:</strong></td>
<td>801-626-7345</td>
</tr>
<tr>
<td><strong>E-Mail:</strong></td>
<td><a href="mailto:dabatwsu@gmail.com">dabatwsu@gmail.com</a> (class questions, read frequently)</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:dbrinkerhoff@weber.edu">dbrinkerhoff@weber.edu</a> (general, read infrequently)</td>
</tr>
<tr>
<td><strong>Time and Room:</strong></td>
<td>Online</td>
</tr>
<tr>
<td><strong>Text:</strong></td>
<td><em>Object-Oriented Programming in C++</em></td>
</tr>
<tr>
<td><strong>Authors:</strong></td>
<td>Robert Lafore</td>
</tr>
<tr>
<td><strong>Prerequisites:</strong></td>
<td>CS 1400, Strongly Recommended: Math 1080 (or equivalent)</td>
</tr>
</tbody>
</table>
| **Grading:** | Programs: 30%  
Midterms: 30%  
Programming Tests: 30%  
Worksheets: 10%  
| | 90% ≤ total < 100% A  
86% ≤ total < 90% A-  
82% ≤ total < 86% B+  
78% ≤ total < 82% B  
74% ≤ total < 78% B-  
70% ≤ total < 74% C+  
62% ≤ total < 70% C  
0 ≤ total < 62% E |
| **Reading:** | Reading assignments are posted on the Canvas Calendar and "Syllabus". |
| **Programs:** |  
- Programming assignments are posted on Canvas as they become available.  
- Programs are submitted via Canvas and are due at 11:59 p.m. on the due date.  
- Please note that there are no group or team programming assignments - all programming assignments are individual. You are encouraged to discuss ideas and problems, and help one another debug programs, but each programming assignment must be your own work.  
- See "Assignments" on Canvas for program details and due dates. |
| **Midterms:** |  
- Please see Canvas for additional details about each midterm.  
- Midterms are delivered on Chi Tester.  
- Midterms are closed book and closed notes but are not timed.  
- Students in the Weber and Davis areas must take the exams in a WSU Testing Center. Students residing outside of this area must provide a proctor at their
own expense. (Students who live in the Salt Lake City area may use the library, which may provide this service at no cost.)

- Students are responsible for knowing the exam dates and the testing center hours and policies.
- Midterm #1: Wed 1/23 - Wed 1/23, 11:59 p.m. [chap 1-3 + pp. 634-640]
- Midterm #2: Wed 2/20 - Wed 2/27, 11:59 p.m. [chap 4-7 + pp. 430-448, 452-457, 641]
- Midterm #3: Wed 3/20 - Wed 3/27, 11:59 p.m. [chap 6, 8-10 + pp. 520-528, 645-647]
- Midterm #4: Wed 4/17 - Wed 4/24, 11:59 p.m. [chap 11-12, 14]

| Programming Tests: | • Please see Canvas for additional details about each programming test.  
• Programming tests consist of a small problem that you solve by writing a program.  
• Programming tests are open book, open notes, and open examples.  
• Programming tests are timed at 60 minutes. This is enough time to solve the problem and write the program but only if you have studied for the test. The test is timed so you can show me what you have learned, not what you can learn.  
• Programming tests may be taken at home or anywhere you have a reliable Internet connection and an ANSI C++ compiler.  
• Programming test are graded with MS Visual Studio.  
• Programming Test 1: Fri 1/18 - Fri 1/25 [Chapter 2]  
• Programming Test 2: Fri 2/22 - Fri 3/1 [Chapters 4 & 5]  
• Programming Test 3: Fri 3/22 - Fri 3/29 [Chapters 6, 8, & 9]  
• Programming Test 4: Fri 4/5 - Thur 4/12 [Chapter 11] |

| Worksheets: | • Worksheets consist of several multiple choice or essay questions.  
• The questions appearing on the worksheets are similar to the questions that will appear on the midterms and are intended to help you prepare for the midterms.  
• Unlike the programming assignments, you may work together on the worksheets.  
• Once you have answered the worksheet questions, you will transfer your answers to ChiTester for scoring. Please note that you cannot stop and restart your ChiTester session - you must transfer all of your answers during one ChiTester session.  
• See "Assignments" on Canvas for worksheet questions and due dates. |

| Incomplete Grades: | An “Incomplete” may be given only when the student, having satisfactorily completed approximately 80% of the required work, is unable to complete the class work for a legitimate reason (such as illness or accident) and can reasonably finish on his/her own. |

| Study Time: | A common “rule of thumb” for allocating study time is 2 to 3 hours of study per credit hour (i.e., 8 - 12 hours per week for CS 1410) in addition to classroom time. This implies that you should plan on 12 to 16 hours of study per week. Please do not overload your schedule. |
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.

Disclaimer: This syllabus is subject to change at any time. Alterations made in class or on Canvas supersede old content and become effective immediately.

Honesty and Fair Use: Each student is expected to maintain high standards of honesty and ethical behavior. Each assignment must represent the student’s own, best effort. You are encouraged to study together and to work together on the labs. This means that it is okay to discuss algorithms, syntax, and C++ in general with others. You may also get ideas and code fragments from books or from the Internet. However, you may not copy whole functions, classes, files, or programs from someone else, from a book or from the Internet; nor may you exchange or share code in any format - including code from previous semesters. If you have any questions about what is acceptable and what is not acceptable, you may ask the instructor.

Please do not ask students (passed or present) to see copies of their assignments. Please do not give other students (current or future) copies of your assignments.

CS Department policy dictates that any verifiable evidence of student academic cheating, as defined above, 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. In the case of one student providing code or other assessment material to another student, these sanctions will apply to both parties. The university may expel students if continued plagiarism is proven. The University affords you certain rights, including the right to challenge the accusation of cheating. The Dean of Students or the Department Chair can explain these rights to you if you are accused of cheating.

Instructor’s Goal: I want to help you learn how to solve problems and how to code the solution in C++. This entails several sub-goals. I want to help you understand: (a) how the object-model works; (b) how to solve problems before you attempt to code the solution in any language; (c) how to debug a program; and (d) how to build appropriate mental models of computer/language systems that will help you understand the tasks on which computer scientists work.

Objectives and Outcomes: At the conclusion of the course you will:

1. Understand and be able to use the fundamental concepts of the ANSI C++ language
   a. variables
   b. expressions
   c. statements
   d. flow-of-control statements (sequential, branching, looping)
   e. functions (definitions and calls), friend functions, virtual functions
   f. structures and unions
   g. pointers, references, and the associated operators and syntax
   h. C++ i/o streams
   i. templates and exception
2. Understand the Object-Oriented model and its relationship to and implementation in the C++ programming language; specifically, you will understand and be able to use:
   a. classes, objects, instantiation (both static and dynamic) and object deallocation
   b. relationships: inheritance, association, aggregation, composition, and dependency
   c. attributes and functions, and their relationship to encapsulation and abstraction
   d. member access and the associated operators and syntax
   e. virtual or abstract classes
   f. polymorphism
3. Understand the physical organization of C++ programs including the organization of multi-class programs
4. Have gained experience solving problems and then expressing the solution to those problems as computer programs