CS 1410 Fall 2015
Object-Oriented Programming Using C++

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

<table>
<thead>
<tr>
<th>Instructor</th>
<th>Delroy A. Brinkerhoff, Ph.D.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office Hours</td>
<td>9:30a-10:30a TTh, 8:30a-11:30a W</td>
</tr>
<tr>
<td>Office Location</td>
<td>TE 111A</td>
</tr>
<tr>
<td>Phone</td>
<td>801-626-7345</td>
</tr>
<tr>
<td>E-Mail</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 questions, read less frequently)</td>
</tr>
<tr>
<td>Time and Room</td>
<td>7:30 a.m. - 9:20 a.m. TTh, TE 103C, and Online</td>
</tr>
<tr>
<td>Text</td>
<td><a href="http://icarus.cs.weber.edu/~dab/CS1410Text.html">http://icarus.cs.weber.edu/~dab/CS1410Text.html</a></td>
</tr>
<tr>
<td>Prerequisites</td>
<td>CS 1400, Recommended: Math 1080 (or equivalent)</td>
</tr>
</tbody>
</table>

Grading:
- Programs: 25% (learning and assessment)
- Midterms: 45% (assessment only)
- Programming Tests: 25% (learning and assessment)
- Worksheets: 5% (primarily learning)

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>92% ≤ total &lt; 100%</td>
</tr>
<tr>
<td>A-</td>
<td>88% ≤ total &lt; 92%</td>
</tr>
<tr>
<td>B+</td>
<td>84% ≤ total &lt; 88%</td>
</tr>
<tr>
<td>B</td>
<td>80% ≤ total &lt; 84%</td>
</tr>
<tr>
<td>B-</td>
<td>76% ≤ total &lt; 80%</td>
</tr>
<tr>
<td>C+</td>
<td>72% ≤ total &lt; 76%</td>
</tr>
<tr>
<td>C</td>
<td>64% ≤ total &lt; 72%</td>
</tr>
<tr>
<td>E</td>
<td>0 ≤ total &lt; 64%</td>
</tr>
</tbody>
</table>

Reading: Reading assignments are posted on the Canvas Calendar.

Assignment Schedules: Assignment due dates (worksheets, labs, tests, and exams) are posted on the Canvas Calendar.

Programming Labs: Please see General Lab Guidelines for details. See the Canvas Calendar for dates.

Midterms: Please see General Midterm Guidelines for details. See the Canvas Calendar for dates.

Programming Tests: Please see General Programming Test Guidelines for details. See the Canvas Calendar for dates.

Worksheets: Please see General Worksheet Guidelines for details. See the Canvas Calendar for dates.

Extra Credit: No extra credit beyond that already specified in the assignments is available.

Late Homework: Late homework is generally not accepted. However, in the case of illness and similar emergencies, accommodations can be made for exams, programming tests, and labs if the instructor is notified before solutions are posted.
**Submitting Homework Via E-Mail**
Homework must be submitted on the Canvas or on ChiTester. I do not accept homework through e-mail or through Canvas messages.

**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.

**Course Completion Options:**
You must demonstrate competency in the use of C++ and the object-oriented software paradigm to pass CS 1410. Based on the four assessment instruments, you may demonstrate competency in one of two ways:

1. Completing the scheduled assessments in all four categories listed above on time and earning a minimum aggregate score.
2. Completing a comprehensive final exam and an extended programming test.

Path 1 represents the normal or traditional way to take a course; path 2 is available for students who:

- Are unable to submit a significant number of assignments due to work, illness, military service, etc.
- Feel that the grade earned through path 1 does not reflect their understanding or capabilities. You must inform me that you wish to take this path before I submit final grades.

**Path 2 students are given a maximum extension of 2 months and must contact the instructor before final grades are submitted.**

**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 worksheets. When working on the labs, you may also discuss algorithms, syntax, concepts, and problems, but you must write your own code.

At your current level of experience and regarding the problems that you will be solving, there is very little of value for you on the Internet. In general you should avoid the Internet and focus on the material provided on Canvas, which focuses on the assignments on which you will be working. It is difficult for me to score your labs when you copy code from the Internet and anytime that you do so, you run the risk of...
failing an assignment or the class. Unless you are explicitly directed to use the Internet, all necessary information is given in or with the lab assignment.

Specifically, 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.

You are guilty of cheating whenever you submit code that you did not write or that was not explicitly provided to you by the instructor.

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
   j. the STL

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