

---

<b>Instructor</b>	Dr. Mark B. Motl
<b>E-mail</b>	Mark.Motl@angelo.edu
<b>Office</b>	MCS 205M
<b>Phone</b>	Voice: 486-5420; FAX: 942-2213
<b>Office Hours</b>	MTWRF 8:30 a.m.– 9:30 a.m. MTWRF 1:00 p.m.– 2:00 p.m.
<b>Web Page</b>	www.cs.angelo.edu/~mmotl/

---

**Objectives** Study of basic data structures and their applications such as: linear structures (arrays, lists, stacks, queues) and non-linear structures (trees, graphs); sequential and linked storage representation methods; sorting and searching algorithms; and techniques of algorithmic analysis.

---

**Prerequisite** CS 1337 (C or better)

---

**Textbooks** Gaddis, Tony, *Starting Out With C++: From Control Structures Through Objects*, Ninth ed., Boston, MA: Pearson Education, Inc., 2018.

*Optional:* Deitel, Paul and Deitel, Harvey, *C++: How to Program*, Ninth ed., Pearson/Prentice Hall, 2014.

---

**Grading** Your course grade will be based on the following:

Assignments	30%
Exams	70%

Overall final average 90%+ = A, 80%+ = B, 70%+ = C, 60%+ = D, <60% = F.

---

**Assignments** The following guidelines are to be observed concerning the programming assignments (labs):

- Labs are due by the end of the day on the designated due date.
- Late assignments will be penalized 15% for each calendar day past the due date.
- The labs will be developed on the department's Unix environment.
- All assignments will be submitted electronically, as discussed in class.
- Assignments will be graded on the CS department's Unix environment. If you develop solutions to the programming assignments in a different environment such as Microsoft Visual Studio, it is your responsibility to ensure that these solutions work correctly in the Unix environment also.
- You are responsible for doing your own work. You may be asked to defend/explain your work at any time. You are encouraged to work with your classmates and use the resources of the Internet to understand and complete the assignments; however, when you submit an assignment, you are verifying that it is your own work. Cheating will not be tolerated.

---

**Exams** Five exams will be administered during the semester. The lowest exam grade will be dropped. There are no provisions for make-up exams. All exams are comprehensive. Parts may be given in the computer lab. Class examinations will be announced at least two classes prior to the examination.

---

<b>Attendance</b>	Attendance is expected, but it will not be used in calculating your final grade.
-------------------	--

---

<b>Academic Honor Code</b>	<p>Angelo State University expects its students to maintain complete honesty and integrity in their academic pursuits. Students are responsible for understanding the Academic Honor Code, which is available on the web at</p> <p style="text-align: center;"><a href="http://www.angelo.edu/forms/pdf/honorcode5.pdf">http://www.angelo.edu/forms/pdf/honorcode5.pdf</a></p>
----------------------------	--

---

<b>Academic Accommodations</b>	Persons with disabilities which may warrant academic accommodations must contact the Student Services Office, Suite 112, Houston Harte University Center, in order to request such accommodations prior to any accommodations being implemented. You are encouraged to make this request early in the semester so that appropriate arrangements can be made.
--------------------------------	--

---

<b>Student Absence for Observance of Religious Holy Day</b>	<ol style="list-style-type: none"> <li>1) "Religious holy day" means a holy day observed by a religion whose places of worship are exempt from property taxation under Texas Tax Code §11.20.</li> <li>2) A student who intends to observe a religious holy day should make that intention known in writing to the instructor prior to the absence. A student who is absent from classes for the observance of a religious holy day shall be allowed to take an examination or complete an assignment scheduled for that day within a reasonable time after the absence.</li> <li>3) A student who is excused under section 2 may not be penalized for the absence; however, the instructor may respond appropriately if the student fails to complete the assignment satisfactorily.</li> </ol>
---	--

---

<b>Student Learning Outcomes</b>	<p>By completing this course, students should:</p> <ol style="list-style-type: none"> <li>1) have a better understanding of the C++ class concept,</li> <li>2) learn techniques of algorithm analysis,</li> <li>3) learn about recursion,</li> <li>4) know how to use the C++ Standard Template Library (STL) <code>vector</code> container,</li> <li>5) learn programming techniques for sorting,</li> <li>6) learn programming techniques for searching,</li> <li>7) know how to use the STL <code>stack</code> adaptor,</li> <li>8) know how to use the STL <code>queue</code> adaptor,</li> <li>9) know how to use the STL <code>deque</code> container,</li> <li>10) know how to use the STL <code>priority_queue</code> adaptor, and</li> <li>11) know how to use the STL <code>list</code> container.</li> </ol>
----------------------------------	---

---

<b>Assessment of Student Learning Outcomes</b>	<p>Methods of assessment:</p> <ol style="list-style-type: none"> <li>1) Programming assignments</li> <li>2) Exams</li> <li>3) Course exit survey</li> </ol>
--	---

---

<b>Course Topics</b>	<p>The following list of course topics is tentative and subject to change and adaptation.</p> <table border="0" style="width: 100%;"> <tr> <td>1) Classes and Data Abstraction</td> <td>7) Algorithm Analysis</td> <td>13) Queues</td> </tr> <tr> <td>2) Program Structure</td> <td>8) Vectors</td> <td>14) Deques</td> </tr> <tr> <td>3) Operator Overloading</td> <td>9) Sorting</td> <td>15) Priority Queues</td> </tr> <tr> <td>4) Recursion</td> <td>10) Searching</td> <td>16) Lists</td> </tr> <tr> <td>5) Template Functions</td> <td>11) Matrices</td> <td></td> </tr> <tr> <td>6) Template Classes</td> <td>12) Stacks</td> <td></td> </tr> </table>	1) Classes and Data Abstraction	7) Algorithm Analysis	13) Queues	2) Program Structure	8) Vectors	14) Deques	3) Operator Overloading	9) Sorting	15) Priority Queues	4) Recursion	10) Searching	16) Lists	5) Template Functions	11) Matrices		6) Template Classes	12) Stacks	
1) Classes and Data Abstraction	7) Algorithm Analysis	13) Queues																	
2) Program Structure	8) Vectors	14) Deques																	
3) Operator Overloading	9) Sorting	15) Priority Queues																	
4) Recursion	10) Searching	16) Lists																	
5) Template Functions	11) Matrices																		
6) Template Classes	12) Stacks																		

---