Catalog Description:

3 hours credit. Review of C/C++ non-object oriented programming concepts. Object-oriented programming including data abstraction, inheritance, operator overloading and polymorphism. Application of OOP to study various data structures including stacks, queues, linked lists, trees and binary trees.

Prerequisites:

CS 2073, EE 3463

Major Prerequisites by Topic:

C Programming, microprocessor programming

Course Objectives:

The objectives of the course are:
1. To learn Object Oriented Programming through C++
2. To apply development tools and human interfaces to software systems using a development environment
3. To introduce algorithms and data structures for application in engineering software problems
4. To provide training in software algorithmic skills and to help students learn software system development skills

Overall, it is expected that the student will learn the skills, develop the motivation, and understand the effort needed to successfully develop engineering-oriented software.

Relationship to EE Department Objectives and Outcomes:

The course objectives primarily address the EE department program outcomes.

(2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
Topics:

1. C++ Programming Language Overview
2. Object Oriented Programming (OOP) concepts such as Data Abstraction, Encapsulation, Inheritance, etc.
3. C++ constructs for OOP
4. Data Structures using C++
   a. Elementary data structures: Lists, Stacks, Queues, etc.
   b. Advanced data structures: Binary search trees, Hashing, etc.
   c. Graphs, and possibly more advanced topics (depending on time)

Performance Criteria:

Course objectives 1 through 4 will be evaluated using evaluation methods [1 - 4]

Evaluation Methods:

1. Two exams during the semester
2. One final exam
3. Assignments
4. Computer programming project

Course Content:

Engineering Science: 2 credits (67%)
Engineering Design: 1 credits (33%)

Class/Laboratory Schedule:

2 hours and 30 minutes of lecture per week. 50 mins of recitation per week.

Coordinator:

Ram Krishnan – Associate Professor of Electrical Engineering
Part B – General Course Information and Policies

Instructor:

Ram Krishnan (http://engineering.utsa.edu/~krishnan/)
Microsoft President’s Endowed Associate Professor
Department of Electrical and Computer Engineering
University of Texas at San Antonio
Email: ram.krishnan@utsa.edu

Teaching Assistant:

Name: Muhammad Rashedul Haq Rashed
Email: rashed.eee09@gmail.com

Lecture time:

Tuesdays and Thursdays, 4pm – 5:15pm @ MH 3.02.18

Recitation time:

Wednesdays, 3pm – 3:50pm @ MH 3.04.26

Office hours:

Tuesdays 11:00am – 1:00pm @ BSE 1.518

Course website:


Supporting materials for this course will be distributed through the website stated above and/or Blackboard.

Reference textbooks:

1. Programming: Principles and Practice Using C++ by Bjarne Stroustrup, Publisher: Addison-Wesley Professional; 1st or 2nd edition
2. Data Structures and Algorithms by Aho, A., Hopcroft, J., and Ullman, J.
4. The C++ Programming Language by Bjarne Stroustrup
5. Internet Resources
Grading:

A letter grade will be determined based on the nature of students’ course performance curve.

Evaluation methods:

1. Exams (2) – 20% + 20%
2. Final – 20%
3. Assignments – 20%
4. Computer programming project – 20%

Attendance:

No penalties will be incurred for absences during regular class hours. However, it is your responsibility to talk to your classmates and keep abreast of topics covered, announcements and assignments given during missed classes. This is a fun course! It’s hands on. Try your level best to attend each lecture!

Late submission policy for assignments:

Each assignment is due at a pre-specified time. Late submissions may receive 50% credit if submitted within one week after the deadline. Assignments submitted one week after the deadline will receive no credit. No exceptions to this policy.

Exam policy:

All exams will be held in-class, closed-book and closed-notes. Exams will have programming component(s). To be fair to all students, there will be no makeup exams. No credit will be given for a missed exam except under extenuating circumstances.

Course evaluation:

Each student completing this course is highly encouraged to evaluate the course toward the end of the semester. The evaluation is used for 2 major purposes: (1) The instructor strongly takes the feedback into account to improve his teaching, and (2) The university utilizes the feedback as one measure to evaluate instructor effectiveness. To encourage students, the instructor offers a 1% extra-credit for completing their course evaluation.

Counseling services, Student code of conduct, Scholastic dishonesty, etc.:

Please visit this webpage: http://utsa.edu/syllabus. Scholastic dishonesty will be taken very seriously.
Tentative Schedule:

Please take a look at the Fall academic calendar and the Fall final exam schedules in UTSA ASAP.

We will meet 30 times in total, out of which, we will have 27 lectures and 3 exams (2 mid-terms and 1 final). We will plan to have approximately 10 assignments.

**Part I: C++ programming basics: 9 lectures**

*Expected date of Exam 1: 09/26*

<table>
<thead>
<tr>
<th>Meeting #</th>
<th>Date</th>
<th>Topics Covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Aug 27</td>
<td>Syllabus overview, Compiling and linking a C++ program</td>
</tr>
<tr>
<td>2</td>
<td>Aug 29</td>
<td>C++ data types, variables, basic programming constructs such as if-else, for/while loop, case, etc.</td>
</tr>
<tr>
<td>3</td>
<td>Sep 03</td>
<td>Intro to pointers</td>
</tr>
<tr>
<td>4</td>
<td>Sep 05</td>
<td>Arrays, pointers and arrays, intro to functions</td>
</tr>
<tr>
<td>5</td>
<td>Sep 10</td>
<td>Creating &amp; using libraries, functions: pass by value vs pass by reference</td>
</tr>
<tr>
<td>6</td>
<td>Sep 12</td>
<td>Arrays and functions, and pointer arithmetic</td>
</tr>
<tr>
<td>7</td>
<td>Sep 17</td>
<td>Intro to dynamic memory allocation</td>
</tr>
<tr>
<td>8</td>
<td>Sep 19</td>
<td>Vectors and implementing Vector-like features using dynamic memory management (dynamic arrays)</td>
</tr>
<tr>
<td>9</td>
<td>Sep 24</td>
<td>Exam 1 review</td>
</tr>
<tr>
<td>10</td>
<td>Sep 26</td>
<td>Exam 1</td>
</tr>
</tbody>
</table>

**Part II: Designing abstract data types in C++ and intro to data structures: 9 lectures**

*Expected date of Exam 2: 10/31*

<table>
<thead>
<tr>
<th>Meeting #</th>
<th>Date</th>
<th>Topics Covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Oct 01</td>
<td>Abstract Data Types (ADT)</td>
</tr>
<tr>
<td>12</td>
<td>Oct 03</td>
<td>C++ features that facilitate ADT: Class, constructors and destructors</td>
</tr>
<tr>
<td>13</td>
<td>Oct 08</td>
<td>Exam 1 solution review</td>
</tr>
<tr>
<td>14</td>
<td>Oct 10</td>
<td>List ADT using static array</td>
</tr>
<tr>
<td>15</td>
<td>Oct 15</td>
<td>List ADT using dynamic array</td>
</tr>
<tr>
<td>16</td>
<td>Oct 17</td>
<td>Singly Linked List, Doubly Linked List</td>
</tr>
<tr>
<td>17</td>
<td>Oct 22</td>
<td>Stacks and Queues</td>
</tr>
<tr>
<td>18</td>
<td>Oct 24</td>
<td>C++ features that facilitate ADT: copy constructors, etc.</td>
</tr>
<tr>
<td>19</td>
<td>Oct 29</td>
<td>Exam 2 review. Mock exam on programming component.</td>
</tr>
<tr>
<td>20</td>
<td>Oct 31</td>
<td>Exam 2</td>
</tr>
</tbody>
</table>
### Part III: Data structures: 9 lectures

**Expected Project announcement:** 11/05  
**Expected Project deadline:** 12/05  

**Final Exam:** Monday, 12/9 @ 3:15 PM

<table>
<thead>
<tr>
<th>Lecture #</th>
<th>Date</th>
<th>Topics Covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>Nov 05</td>
<td>C++ features that facilitate ADT: operator overloading, etc.</td>
</tr>
<tr>
<td>22</td>
<td>Nov 07</td>
<td>Misc C++ features: Const keyword, this pointer, reference variables</td>
</tr>
<tr>
<td>23</td>
<td>Nov 12</td>
<td>Exam 2 solution review, Binary Search Tree (BST) – add element</td>
</tr>
<tr>
<td>24</td>
<td>Nov 14</td>
<td>BST Traversal (preorder, inorder, postorder), Overloading &lt;&lt; operator</td>
</tr>
<tr>
<td>25</td>
<td>Nov 19</td>
<td>BST – delete element</td>
</tr>
<tr>
<td>26</td>
<td>Nov 21</td>
<td>BST – delete element</td>
</tr>
<tr>
<td>27</td>
<td>Nov 26</td>
<td>Attend Tech Symposium. No classes in College of Engineering.</td>
</tr>
<tr>
<td>28</td>
<td>Nov 28</td>
<td>Thanksgiving recess (classes do not meet)</td>
</tr>
<tr>
<td>28</td>
<td>Dec 03</td>
<td>Open and Closed Hashing</td>
</tr>
<tr>
<td>29</td>
<td>Dec 05</td>
<td>Final review</td>
</tr>
<tr>
<td>30</td>
<td>Dec 09</td>
<td>Final Exam @ 3:15pm</td>
</tr>
<tr>
<td></td>
<td>Dec 16</td>
<td>Grades due by 2pm</td>
</tr>
</tbody>
</table>
