ECE 4050 Syllabus, Fall 2011

No: ECE 4050
Title: Algorithms and Data Structures
Credits: 4

WSU Catalog Description:
Prereq: Knowledge of C or C++ programming through arrays and pointers. You should have completed the programming class with a C grade or above.
Introduction to problem solving methods and algorithm development, data abstraction for structures such as stack, queues, linked lists, trees, and graphs, searching and sorting algorithms and their analysis.

Instructor: Brady King, Research Assistant at Children’s Hospital of Michigan
Office Hours: T/Th 6:00 – 7:20
Office: 3157 Engineering
Phone: 248-217-4130
Email: bwking@wayne.edu
Course Meeting Time: T/Th 7:30 – 9:20
Course Meeting Location: MANO 279

Course Website: We will use Blackboard (You are responsible for all announcements on the course homepage).

Goals: To introduce the students to problem solving methods and algorithm development and data abstraction that serve as the basis of complex algorithms and applications.

Learning Objectives: Upon successful completion of this course the student will be able to

- use data structures and algorithms in C++ programs
- recognize multiple methods for implementing a solution to a problem and designate one as the most efficient
- define a fundamental abstract data type and discuss its various implementations
- analyze algorithms
- implement generic data structures using templates
- implement fundamental algorithms (e.g. searching, sorting, hashing, traversal algorithms, etc.)
- use UML class diagram notation to describe the interface of classes
- apply theoretical knowledge about algorithms to solve engineering problems using a computer

C++ How to Program, Harvey Deitel and Paul Deitel, 5th Edition

Prerequisites by Topic: Knowledge of C or C++ programming through arrays and pointers. You should have completed the programming class with a C grade or above.
Corequisites by Topic: None

Topics:

- A thorough review of C/C++ programming, including stream I/O, loops, functions, structs, arrays, pointers, and dynamic memory allocation
- Object-oriented design including encapsulation and information-hiding, separation of behavior and implementation, inheritance, operator overloading, templates, polymorphism, exception, and UML class notation
- Data abstraction using object-oriented programming techniques
- Algorithms and problem-solving: problem-solving strategies, the role of algorithms in the problem-solving process, implementation strategies for algorithms, debugging strategies, the concept and properties of algorithms
- Basic searching and sorting algorithms
  - Sorting: selection, insertion, bubble, heap, radix, quick and merge sorting techniques
  - Searching: linear search, binary search and basic searching structures
- Recursion and recursive algorithms
- Implementation of the fundamental abstract data types using pointers, arrays and templates:
  - Linear data structures such as lists, stacks, queues, deques, and sets
  - Hierarchical data structures such as binary trees and ordered oriented (or general) trees
  - Search structures such as hash tables, binary search trees, balanced trees, priority queue implementations
  - Algorithms that make use of these data structures
- Basic algorithm analysis

Course Structure: The class meets for two lectures a week (T/Th 7:30 – 9:20) in MANO 279

Computer Recourses: Each student should have access to a computer for programming assignments

Course Resources: None

Distribution of Points:

- Midterm Exam 20%
- Final Exam 30%
- Programming Assignments 40%
- Attendance and Participation 10%

Grading Scale:

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Grade</th>
<th>Honor Point Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>93-100</td>
<td>A</td>
<td>4.00</td>
</tr>
<tr>
<td>90-92</td>
<td>A-</td>
<td>3.67</td>
</tr>
<tr>
<td>87-89</td>
<td>B+</td>
<td>3.33</td>
</tr>
<tr>
<td>83-86</td>
<td>B</td>
<td>3.00</td>
</tr>
</tbody>
</table>
Attendance: Every student is expected to attend all lectures

Exams:

- The midterm exam will be scheduled based on the class progress and will be announced at least one week in advance.
- The final exam is scheduled for Wednesday December 15th during normal class time. You must let me know during the first week of classes if you have a conflict with another exam or you have exam overload.
- Do NOT assume that you will be able to take a makeup exam. Regardless of your circumstances, approval to take a makeup exam is up to the instructor.
- All exams are closed-book, closed-notes except for only one single-sided 8 ½” x 11” sheet of hand-written notes.
- Calculators that store notes are not allowed.
- Flagrant cheating on an exam will result in, at minimum, a failing grade for the course.

Assignments:

- Assignments will be posted on the course webpage (Blackboard).
- Assignments must represent your original work. You must not look at other solutions or show your solutions to anyone else. At minimum, duplicate or very similar assignments will receive negative grades.
- Save all intermediate work until an assignment has been graded, returned, and recorded. Keep the final source version of your programs. Make frequent backups of your work. Never let anyone else use your account.
- All questions on grading must be brought to my attention within one week of the assignment's return.
- You are welcome to ask any questions with regard to the assignments after class or during office hours. You do not have to have an appointment to come to the office hours.
- You should always start working on the assignments as soon as they are announced, even if they seem to be easy to you. You may run into unexpected problems which you may not be able to solve on your own when it is too late for you to ask for help.

Late-Submission Policy:
- Late assignments will NOT be accepted. The following excuses will NOT be approved for late submissions: computer crashes, disk crashes, accidental file deletions, lab computer unavailability, forgetting to print out the checklist and/or the output, printer problems, and the like.
- You are strongly encouraged to turn in the assignments in the class before the deadline to account for any unpredictable situations. You must always work ahead and make backups to account for unexpected problems.

**Cheating Policy and Penalty for Cheating:** Cheating is defined by the university as “intentionally using or attempting to use, or intentionally providing or attempting to provide, unauthorized materials, information, or assistance in any academic exercise.” This includes any group efforts on assignments or exams unless specifically approved by the professor for that assignment/exam. Evidence of fabrication or plagiarism, as defined by the university in its brochure Academic Integrity, will also result in downgrading for the course. Students who cheat on any assignment or during any examination will be assigned a failing grade for the course.

**Outcome Coverage:**

(a) An ability to apply math, science and engineering knowledge. The homework and exams require students to solve problems using C++ and data structures techniques.
(b) An ability to design and conduct experiments, as well as to analyze and interpret data. The homework and project assignments require the student to analyze, design and implement applications using C++ applications and data structures.
(c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. The design and implementation of the project must be checked against real world constraints.
(e) An ability to identify, formulate and solve engineering problems. Students must be able to design and develop C++ applications and predict their performance under practical limits.
(f) An understanding of professional and ethical responsibility. Students will learn how not to misuse or abuse their knowledge in programming.
(g) An ability to communicate effectively. Students are required to write documentation on their project.
(h) The broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context. The course details the design of engineering solutions to meet global, economic, environmental, and societal needs.
(j) A knowledge of contemporary issues. The students will learn about contemporary issues with C++ programs and how these issues are being addressed.
(k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. Students taking the course will learn how to use software tools such as Visual Studio .NET and the UNIX operating system along with the C++ programming language to build applications.