Course Homepage:
http://www.ece.eng.wayne.edu/~sjiang/ECE4050-winter-16/ECE4050.htm

Course Description

This course is designed to provide students with an understanding of problem solving methods and algorithm developments and analysis skills. Students will learn data abstraction for structures such as lists, stacks, queues, binary and non-binary trees, as well as various searching and sorting algorithms.

Topics:

General concepts of data structures and algorithms, preliminaries of algorithm analysis, and data structures (lists, stacks, queues, binary/non-binary trees, sorting, searching, and indexing algorithms).

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

- Understand various data structures and perform common operations on them.
- Understand and employ commonly used algorithms to design larger-scale programs.
- Understand various constraints in the design of an algorithm for a problem according to given design goals and available resources.
- Select the most appropriate data structure and algorithm among a number of them for optimal efficiency or performance.
- Mathematically analyze efficiency and performance of algorithms.
- Identify performance bottlenecks of an existent algorithm or its implementation and optimize them by alleviating the bottlenecks.
- Use C or C++ to implement algorithms and conduct real-system performance measurements and analysis

Outcome Coverage:

(a) an ability to apply math, science and engineering knowledge. The homework and exams require students to solve problems in area such as asymptotic algorithm analysis and program coding and running.

(b) an ability to design and conduct experiments, as well as to analyze and interpret data. The homework and project assignments require students to design and implement various data structures and algorithms, as well as debugging and tuning programs on computers. Performance evaluation and analysis are required in some of the projects.

(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 in the project must be checked against real
world operating limits, such as program running time and memory space constraints.

(d) an ability to identify, formulate and solve engineering problems. Students must be able to identify disadvantages of a data structure or a algorithm and propose solutions to improve them.

(e) an understanding of professional and ethical responsibility. Students are required to design algorithms that allow computer to be most efficiently operated in terms of performance and energy consumption.

(f) an ability to communicate effectively. Students are encouraged to ask questions, be involved in in-class discussions and problem solving.

(g) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context. The course covers application of data structures and algorithms in today’s data centers and large-scale Internet-wide service, as well as their society impacts.

(h) knowledge of contemporary issues. The students learn the design and issues with data structures and algorithms that are being used by millions of people and what issues are being addressed.

(i) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. Students are required to use C++ software development tools to conduct many programming-intensive projects.

Course Prerequisites

Knowledge of C or C++ programming with arrays and pointers. You should have completed the programming class with a C grade or above.

Course Textbook:


Computer Recourses:

Each student should have access to a computer for programming assignments, preferably with Microsoft Visual Studio installed.
Class Attendance

Class attendance is required. Please arrive on time.

Grading Policy

Distribution of Points:

✓ 30% Programming assignments
✓ 60% Exams
  ▪ 15% midterm X 2
  ▪ 30% final
✓ 10% Class attendance

Grading Scale:

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>90-100</td>
<td>A</td>
</tr>
<tr>
<td>85-89</td>
<td>A-</td>
</tr>
<tr>
<td>80-84</td>
<td>B+</td>
</tr>
<tr>
<td>75-79</td>
<td>B</td>
</tr>
<tr>
<td>70-74</td>
<td>B-</td>
</tr>
<tr>
<td>65-69</td>
<td>C+</td>
</tr>
<tr>
<td>60-64</td>
<td>C</td>
</tr>
<tr>
<td>59-55</td>
<td>C-</td>
</tr>
<tr>
<td>&lt; 55</td>
<td>F</td>
</tr>
</tbody>
</table>

Exams:

- The midterm exams will be scheduled based on the class progress and will be announced at least one week in advance.
- The final exam is scheduled during university-wide exam period.
- 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 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.
- 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.
• All questions on grading must be brought to the instructor’s 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, but with the following exception: every student of the course is allowed to have a total of THREE (3) FREE LATE DAYS. Consider them as non-transferable, non-replaceable credits and use them wisely. If you wish to use one or more of these, indicate it clearly in your submission. No partial late day is allowed. You need one whole late day to cover a one-hour late submission. Late submissions will not be graded if you have used all two late days.

Note that 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 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.