From Dr. Yelena Rykalova's home page

COMPII-GtCOMPII: COMP1020

COMP 1020 Spring 2018

Home Syllabus Lecture Blog

Catalog Description

Computing II focuses on the implementation and applications of data structures, including arrays, linked lists, stacks, queues, trees, binary trees, binary search trees, heaps, graphs, and hash tables. Recursive approaches are used. Performance analysis is discussed. Attention is paid to programming style, documentation, and testing. This course includes extensive laboratory work. Effective Fall 2013, Co-req: Computing II Lab.

Course Prerequisites

COMP.1010 Computing I and COMP.1030 Computing I Lab.

Required Textbook

Algorithms in C, Parts 1-5: Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms 3rd edition, Robert Sedgewick

Supplementary Textbook

- Data Structures An Advanced Approach Using C, Esakov and Weiss
- Data Structures and Program Design in C, Second Edition by Kruse, Tondo and Leung.

Required Software

- Microsoft Visual Studio

ABET Student Outcomes

1. An ability to apply knowledge of computing and mathematics appropriate to the discipline.
2. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
3. An ability to use current techniques, skills, and tools necessary for computing practice.

4. An ability to apply design and development principles in the construction of software systems of varying complexity.

Learning Outcomes

By the end of this course, students will be able to:

- Use command line prompt to compile and debug their code
- Develop an understanding about the different types of data structures and how to use them, including linked list, stacks, queues, trees, graphs, and hash tables
- Design and write C programs to implement the different data structures
- Employ testing in their program implementation
- Use valgrind tool to detect memory leaks
- Create Make files

Grading

- 75% of total grade:
  - 50% Programming Assignments
  - 30% 2 Midterm exams
  - 20% Final Exam
  - 5% *Extra points* *Pop Quizzes*

- 25% of total grade:
  - Lab exercises:
    - 60% of Lab grade - Lab 1 - 6
    - 35% of Lab grade - Lab 7 & 8
    - 5% Attendance

      Late penalty - 5% off from the total grade for every Lab submitted late

Course Policies

Lectures

You are responsible for all materials presented in lecture.

Collaboration

- There are no group assignments in this course. The programs are expected to be your own. You may discuss assignments and general approaches to solve a problem with the instructor, Teaching assistants, or your classmates. However, you must design and write your code by yourself. You could ask the instructor or teaching assistants for help in debugging your code.
- You cannot post any solutions for the assignments online.
- Any piece of code that was obtained through other means, e.g., interent or books, must be clearly cited.

- Any attempts to use other classmates' code or failing to properly cite any obtained code will be considered as PLAGIARISM. Check the university **academic integrity rules** for more information.

No Posting of Solution Code Policy

You are not allowed to post solution code to problem sets assigned in this class in public places (e.g. Github). This includes your own solutions as well as solutions that may be provided by the instructors.
This policy is a courtesy to future students, who — to the fullest extent possible — should have the opportunity to struggle with the problems in the same way that you do.
Please note that this is typical policy at premier computer science departments. E.g.:

- *Princeton COS 126*. "Your work must never be shown or communicated to anyone who is taking COS 126 now or who might take COS 126 in the future. ... You must never place your work in any public location (including websites, leaving printouts in a classroom, etc.). ... The rules ... continue to apply even after this semester is over."
- *Harvard CS50*. "Not reasonable: Providing or making available solutions to problem sets to individuals who might take this course in the future."
- *MIT 6.01*. "Students should never share their solutions (or staff solutions) with other students, including through public code repositories such as Github." (emphasis in the original)

Non-compliance will be pursued rigorously per UMass Lowell's academic integrity policy.

Student Affairs Information and Services for Students

- see **Student Affairs Information and Services for Students**

Course Schedule

Tentative Schedule

This is a tentative schedule and subject to change. Please check frequently.

Week	**Topic**	
1 (Jan. 22)	**Introduction, c review, and introduction to software engineering**	
2 (Jan. 29)	**Vectors**	Lab 1
3 (Feb. 5)	**Type Independent Vectors; Linked Lists**	Lab 2 **Lab 1 due**
4 (Feb. 12)	**Linked Lists; Type Independent Stack**	Lab 3 **Lab 2 due**

5 (Feb. 19)	Queues	
6 (Feb. 26)	Recursion, Generic Vectors	Lab 4 Lab 3 due
7 (Mar. 5)	Midterm1 Mar 7 Generic Vectors	
8 (Mar. 12)	Spring Break	
9 (Mar. 19)	Sorting, Intro to Trees Binary Trees, AVL trees	Lab 5 Lab 4 due
10 (Mar. 26)	Searching Toward Building an Associative Array	Lab 6 Lab 5 due
11 (Apr . 2)	Midterm2 AVL tree, Priority Queue, Heap	Lab 7 & 8 Apr 6 due
12 (Apr . 9)	Priority Queue, Heap	
13 (Apr . 16)	Priority Queue, Heap, Hash table	
14 (Apr . 23)	Hash table, Heap, and Priority Queues	Lab 7 & 8 due Fr 27 is the last day to submit
15 (Apr . 30)	Binomial Queue, Heap, and Priority Queues. Final Exam review	

Retrieved from http://www.cs.uml.edu/~yrykalov/pmwiki/pmwiki.php/COMPII-GtCOMPII/COMP1020
Page last modified on January 22, 2018, at 04:46 PM