COP 4530: Data Structures

3 Credits
Fall 2014

- *Administrative Information*

  1. *Instructor:* Dr. Tina Kouri
  2. *Department:* Computer Science and Engineering, College of Engineering
  3. *Office:* ENB 339
  4. *E-Mail:* `tkouri@cse.usf.edu`
  5. *Phone:* (813) 974-2126
  6. *Walk-in Office Hours:*
     - Monday/Wednesday, 12:00pm-2:00pm
     - Tuesday/Thursday, 3:00pm-4:00pm
  7. *Meeting Time and Location:* Tuesday/Thursday, 12:30pm-1:45pm, ENB 118
  8. *Course Web Page:* Canvas (`https://usflearn.instructure.com/`)
  9. *TAs:*
     - Dakun Shen, `dakun@mail.usf.edu`, ENB-327. Walk-in Office Hours: Wednesday, 3:00pm-5:00pm
     - Dmitrii Cherezov, `cherezov@mail.usf.edu`, ENB-329. Walk-in Office Hours: Friday, 1:30pm-3:30pm

- *Approved Course Description*
  Understand and implement fundamentals of concise data structure and organization for program efficiency, clarity and simplification. Implementation of different data types and structures. Understanding of current data structures.

- *Prerequisites*
  COP 3331 (Object Oriented Design) and COT 3100 (Introduction to Discrete Structures)

  - *Sorting:* bubble sort, merge sort, insertion sort
  - *Search:* sequential search, binary search
  - *Proofs:* by contradiction, induction
  - *Programming:* Competent C++ programmer.

- *Course Objectives*

  1. Think in terms of Abstract Data Types (ADTs) and the associated constraints associated in their use and implementation in terms of information hiding.
  2. Become familiar with standard ADTs, such as stacks, queues, linked lists, binary trees, general trees, heaps, graphs, and hash tables.
  3. Be able to implement ADTs in imperative languages.
  4. Be able to implement ADTs using functional programming concepts.
  5. Design and analyze solutions to complex programming tasks using appropriate data structures.
  6. Develop the ability to compare and contrast different implementations of an ADT.
  7. Design, implement and use new ADTs.
  8. Become familiar with concepts of software reuse, documentation, and good programming principles.

- *Required Textbook*
  ADTs, Data Structures and Problem Solving with C++, 2nd edition, Larry Nyhoff (ISBN 0-13-140909-3)

- *Grading*
  - *Grading Rubric*
    1. Exam 1: 10%
       Tentatively scheduled on September 18, 2014, in-class
    2. Exam 2: 10%
       Tentatively scheduled on October 16, 2014, in-class
    3. Exam 3: 10%
       Tentatively scheduled on November 13, 2014, in-class
    4. Comprehensive Final Exam: 20%
       Scheduled on December 11, 2014, 10:00am-12:00pm (in accordance with the USF final exam matrix)
    5. Projects: 50%
    6. Homework (individual): 0%
  - *Grading Scale*
    Your weighted average for the course, using the above rubric, must be greater than or equal to <avg> in order to receive a grade of <grade> (see table below for <avg> and <grade> corresponding values).

<avg>	<grade>
90.0%	A
80.0%	B
70.0%	C
60.0%	D
0%	F

  - The instructor reserves the right to make minor adjustments to the above grading scale based on class averages.
  - The instructor reserves the right to use the +/- grading system.
  - In order to pass the class, a student must be deemed to have a passing total score AND a passing score in the project component of the class AND a passing score in the examination component of the class.[1]
  - Incomplete grades will not be given.

[1] Passing is defined as earning a grade of "C-" or better (weighted average of 70% or higher).

- *Course Policies & Procedures*
  - *Project Policies*
    - Projects are due at the *beginning* of class on the assigned due date. Late projects will be penalized 20% of the possible points for the entire project for each 24 hour period late.
    - Project reports[2] must be prepared with a word processor and submitted as a PDF file to Canvas. Hand-written submissions will not be accepted.
    - Project code must be written in C++ (g++ compiler, version 4.8.2) or Python (version 2.7.6). The programming language(s) to be used will be specified in the project description.
    - Project code must compile and run on the C4 Linux Lab machines. If your code does not compile and run on the C4 Linux Lab machines you will earn 0 points for the project. *If you develop your software in an integrated IDE on another platform, it is your responsibility to test your work on the C4 Linux Lab machines.*
    - You must submit a hard copy and an electronic copy of all projects. If you do not submit both a hard copy and an electronic copy of a project then you will earn 0 points for the project.
  - *Group Project Grading*
    - Grade Computation
      - Let x denote the score assigned, in accordance with the grading rubric, based on the quality of the work submitted by the group.
      - Let y denote your individual effort percentage assigned by the group.
      - Let z denote the number of students in your group.
      - Let w denoted the number of points allotted for the project.

      Your individual grade, g, is computed using the formula: $g = \min(xyz, w)$.
      Individual grades are capped at the number of points allotted for the project.
    - Example Computations:
      A project is worth 50 points (i.e., $w = 50$) and the group earns a quality grade of 40 points (i.e., $x = 40$).
      - Suppose the group assigns effort percentages of 40%, 40%, and 20% to its three members A, B, and C, respectively. The grade computation for each student is:

Student	x	y	z	w	g
A	40	0.4	3	50	$\min(40 \times 0.4 \times 3, 50) = \min(48, 50) = 48$
B	40	0.4	3	50	$\min(40 \times 0.4 \times 3, 50) = \min(48, 50) = 48$
C	40	0.2	3	50	$\min(40 \times 0.2 \times 3, 50) = \min(24, 50) = 24$

      - Suppose the group assigns effort percentages of 50%, 30%, and 20% to its three members A, B, and C, respectively. The grade computation for each student is:

Student	x	y	z	w	g
A	40	0.5	3	50	$\min(40 \times 0.5 \times 3, 50) = \min(60, 50) = 50$
B	40	0.3	3	50	$\min(40 \times 0.3 \times 3, 50) = \min(36, 50) = 36$
C	40	0.2	3	50	$\min(40 \times 0.2 \times 3, 50) = \min(24, 50) = 24$

    - Group assigned percentages must be submitted with the project submission.
    - Group assigned percentages cannot be changed after the project has been submitted.
    - The instructor reserves the right to make adjustments to the group assigned effort percentages if the reported effort percentages (in the instructor's judgment) do not accurately reflect individual contributions.
  - *Homework Policies*
    - Homework is usually assigned the class period of the first lecture for a chapter and is usually due the class period following the last lecture for the chapter.
    - Homework assignments are due at the *beginning* of class on the assigned due date. Late homework assignments are not accepted.
    - Homework assignments are not graded. Homework is assigned to help you learn the material.
    - Homework assignments should be submitted as a PDF file to the designated submission area in Canvas.
  - *Exam Policies*
    - Requests for make-up examinations will not be entertained. I will only make exceptions to this policy in case of excused absences. You must provide sufficient documentation to prove that your absence is excused.
    - You must bring your University of South Florida identification card to each exam. The identification card will be verified during each exam.
    - The dates provided for the exams are tentative. Changes to an exam date will be announced in-class at least one week prior to the exam.

[2] Projects reports are not required for all projects.

- *Regrade Policies*
  - Regrade requests must be submitted, in writing, to the instructor within seven calendar days of either: (1) the date the graded material is returned in class or (2) the date the grades are posted on Canvas, whichever occurs first.
  - Regrade requests must be written on a separate sheet of paper and must be attached to your original submission.
  - Regrade requests must specify the question(s) to be regraded. Regrade requests must include a brief description of why the question(s) should be regraded.
  - The instructor reserves the right to regrade the entire submission.
  - Graded material which has been modified in any way since it was returned to the student will not be regraded.
- *Attendance Policies*
  - Students are expected to attend all classes.
  - You are responsible for all material presented during each lecture. Material presented during the lecture may not be in the textbook.
  - Students who anticipate the necessity of being absent from class due to the observation of a major religious observance must provide notice of the date(s) to the instructor, in writing, by the second class meeting.
- Hand-written work must be legible. If your work is illegible it will not be graded (i.e., you will earn a grade of 0)
- Students in need of academic accommodations for a disability may consult with the office of Students with Disabilities Services (SDS) to arrange appropriate accommodations. Students are required to give reasonable notice prior to requesting an accommodation.
  - If you require extra time on exams due to your disability, you are required to make arrangements to take your exams with the SDS office. You will not receive extra time if you choose to take your exams with the course instructor.
- You may tape my lectures and take notes for personal use, but you may not redistribute the tapes/notes or make monetary profit from the tapes/notes.
- The instructor reserves the right to interpret the class policies and procedures if confusions occur.

- *Academic Integrity/Academic Dishonesty*
  I expect students to be honest and not cheat on their homework/projects/exams. Each student should work independently on all homework, examinations, and individual projects. Students should only work with their assigned group members on group projects. In light of this, I expect you to read the University's policies on student conduct, academic dishonesty, etc. Please see the University's Undergraduate Catalog regarding these policies at `http://www.ugs.usf.edu/catalogs/1213/pdf/AcademicIntegrityOfStudents.pdf`. I also expect you to read and understand the ACM definition of plagiarism (`http://www.acm.org/publications/policies/plagiarism_policy`). Students caught cheating in any form will receive an FF grade.

  The University of South Florida has an account with an automated plagiarism detection service which allows instructors to submit student assignments to be checked for plagiarism. I reserve the right to submit assignments to this detection system. Assignments are compared automatically with a huge database of journal articles, web articles, and previously submitted projects and papers. The instructor receives a report showing exactly how a student's paper or code was plagiarized.