COP 3530 Data Structures and Algorithms

1. **Catalog Description:** *Credits: 4;*

Algorithm development using pseudo languages, basic program structures, program design techniques, storage and manipulation of basic data structures like arrays, stacks, queues, sorting and searching and string processing. Linked linear lists. Trees and multilinked structures. (M)

2. **Pre-requisites and Co-requisites:**

COP3504 or COP 3503 with a minimum grade of C ; COT 3100; and MAC 2234, MAC 2312, MAC 3473, or MAC 3512.

3. **Course Objectives:**

A successful student will

- be able to analyze problems from a computing perspective, propose and evaluate solutions to problems asymptotically using mathematical analysis as well as empirical methods;
- be familiar with and able both to use correctly and to implement efficiently fundamental data structures and algorithms;
- develop, analyze, verify, validate, and test programs against a set of requirements, including preparation of a suitable test plan;
- understand the object-oriented programming paradigm and put it into practice to solve medium-sized programming problems using C++;
- understand the importance of and consistently use data and process abstraction;
- understand the importance of and consistently use good programming practices including good documentation;

This is the cornerstone course for computer science and engineering. It teaches students how to think about and solve problems using algorithms and data structures, and how to realize these abstract notions in computer programs. It is intended to prepare students for serious programming, and as such it is demanding, with several significant projects required. In addition to programming projects, a degree of mathematical sophistication is required to analyze performance and correctness of algorithms and their implementations. It is the gateway course to several other computer science and engineering courses for good reason.

4. **Contribution of course to meeting the professional component:**

Engineering science is addressed in the theoretical aspects of performance analysis, e.g., worst case and average case space and time requirements for algorithm execution on inputs of a given size, and asymptotic analysis. Engineering design and good development practice are important parts of this course. Students will study and evaluate data structures and algorithms abstractly, as well as design, implement, test, and document their own solutions to programming problems, using modern development and testing tools.

5. **Relationship of course to program outcomes:**

Skills student will develop in this course (ABET)

- a) an ability to apply knowledge of mathematics, science, and engineering.
- e) an ability to identify, formulate, and solve engineering problems.
- k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

6. **Instructor:**
   Prof. R. Newman
   a. Office location: CSE-E346
   b. Telephone: 352-505-1579 (not best way to reach me outside of office hours)
   c. E-mail address: nemo-at-cise-dot-ufl-dot-edu
   d. Class Web sites: http://www.cise.ufl.edu/cop3530sp15, http://www.cise.ufl.edu/~nemo/cop3530
   e. Office hours: TBD
7. **Teaching Assistant:**
   Md Muhmudul Hasan
   a. Office hours location: CSE-E309
   b. Telephone: tba
   c. E-mail address: mmhasan at cise dot ufl dot edu
   d. Office hours: **M 10th, T 9th & 10th periods**
   Rahul Prabhu
   e. Office hours location: CSE-E309
   f. Telephone: tba
   g. E-mail address: rprabhu at cise dot ufl dot edu
   Office hours: TBD
8. **Class meeting times:**
   MWF 8th period (3:00-3:50) in PUGH 170
9. **Discussion Section Schedule: (bring your laptop to discussion)**
   Sect 1079: T 8th period TUR L005
   Sect 6723: M 9th period TUR L005
10. **Meeting Location:**
    Class – PUGH 170
    Discussions – TUR 005
11. **Material and Supply Fees: n/a**
12. **Textbooks and Software Required**
    a. Title: Algorithms in C++, Parts 1-5: Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms, 3/E
    b. Author: Robert Sedgewick
    c. Edition: 3rd
    d. ISBN: 0201350882 — Parts 1-4 (Volume 1)

ISBN: 0201361183 — Part 5 (Volume 2)
ISBN: 020172684X — Parts 1-5 (Volumes 1 & 2 bundled)
ISBN: 0321735374 — Parts 1-5 & C++ Backpack Reference Guide (The “super bundle”)

13. **Recommended Reading:**
    a. Title: **C++ PRIMER**
    b. Author: **LIPPMAN, LAJOIE, & MOO**
    c. Publication date and edition: 2013, 5/e, **ADDISON-WESLEY**
    d. ISBN: **0321714113**

    a. Title: **DATA STRUCTURES ALGORITHMS AND APPLICATIONS IN C++**
    b. Author: **SAHNI**
    c. Publication date and edition: 2/e, **SILICON PRESS**
    d. ISBN: 0929306325

2. **Course Outline (topics covered by week, approximately)**
   a. Fundamentals (1 wk) – analysis of algorithms, recurrence relations, big-O

b. Data Structures (4 wks) – arrays, linked lists, memory management, strings, ADTs, stacks, queues, trees, recursion
c. Sorting (3 weeks) – bubblesort, shellsort, quicksort, mergesort, heapsort, radixsort
d. Searching (3 weeks) – symbol tables, BSTs, balanced trees, hashing, radix search, external searching
e. Graphs (5 weeks) – graph types and representations, graph search, digraphs, DAGs, MSTs, shortest paths, network flow

3. **Attendance and Expectations:**

Attendance is required. Pop quizzes may be given on assigned reading and on material covered in classes.
Cell phones and pagers must be silent during class. Reading emails, facebook, etc. is appropriate at some other time and place.
Do bring your laptop with you to respond to on-line questions and to work out programming problems in discussion.
Questions are encouraged - raise your hand to be recognized. Try to formulate the question before asking it, and wait to see if it is answered in a few minutes so we can maintain flow. Lengthy discussions will be deferred to office hours.
Students are required to check the class web pages at least three times a week (MWF) for announcements/updates.
You are responsible for all assignments posted on the web page or announced in class.
Projects are all to be done on an individual basis.

4. **Grading – methods of evaluation:**
   a. Quizzes: 15%
   b. Homeworks: 5%
   c. Exams: 45%
   d. Projects: 35%

Project grades include scoring for documentation and good programming practice in addition to efficiency and correct functionality. Refer to the undergraduate software grading rubric for more information.

5. **Grading Scale:**

Grades are curved. However, there some guidelines you may follow:
a. A = 80% on exams, 90% on projects, 90% on homeworks and quizzes
b. B = 70% on exams, 80% on projects, 80% on homeworks and quizzes
c. C = 60% on exams, 70% on projects, 70% on homeworks and quizzes

Significantly better performance in one method may make up for poorer performance in another method up to a point, but you cannot pass unless you have earned at least 60% on your projects.

Obligatory Statements

"A C- will not be a qualifying grade for critical tracking courses. In order to graduate, students must have an overall GPA and an upper-division GPA of 2.0 or better (C or better). Note: a C- average is equivalent to a GPA of 1.67, and therefore, it does not satisfy this graduation requirement. For more information on grades and grading policies, please visit:
https://catalog.ufl.edu/ugrad/current/regulations/info/grades.aspx
"Undergraduate students, in order to graduate, must have an overall GPA and an upper-division GPA of 2.0 or better (C or better). Note: a C- average is equivalent to a GPA of 1.67, and therefore, it does not satisfy this graduation requirement. Graduate students, in order to graduate, must have an overall GPA of 3.0 or better