CSIS 4104: Data Structures & Algorithms II
Fall 2018 Syllabus

Instructor: Dr. Vincent A. Cicirello
Office: G116
E-mail: Please use course Blackboard mail tool.
Phone (office): x3526

Office Hours: Tuesday/Thursday 10:30am-11:30am
Available other times by appointment; drop-ins also welcome

Course Time and Location: Tuesday/Thursday: 12:30pm-2:20pm, G108

Course Description: In this course, students deepen their knowledge of the design and analysis of computer algorithms. Advanced topics in algorithms and algorithm analysis covered in the course include graphs and graph algorithms, string matching, multi-threaded algorithms, and NP-completeness.

Prerequisites:
- CSIS 3103: Data Structures & Algorithms I (C or better),
- CSIS 2226: Foundations of Computer Science (C or better), and
- MATH 2215: Calculus I (C or better).

This course is a Q2 (Quantitative Reasoning Across the Disciplines): Among the math concepts from prior courses that you will be using in this course are the following: (a) discrete math topics including set theory, logic, graphs, trees, and several other topics covered in MATH 2225/CSIS 2226; and (b) calculus topics including limits and derivatives (you might want to dust off your calculus textbook if you still have it).

Required Textbooks/Readings:
- Introduction to Algorithms, 3rd Edition, 2009, by T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. ISBN: 9780262033848.

Useful/Suggested Resources:
- Your calculus textbook (if you still have it).

Other Required Resources:
- Python 3 (current version if 3.7.0, but any version that starts with 3 is sufficient): https://www.python.org/
  - Recommended version for this course: **WinPython**
    - It's a "portable" distribution (i.e., you can run it from a flash drive, etc)
    - http://winpython.github.io/
    - **Download page:** https://sourceforge.net/projects/winpython/files/ (Look for link where it says "Looking for latest version")
    - **Why?** WinPython can be installed and run from a USB flash drive, so if you do that, then you can work on course work in any lab regardless of whether Python is installed in that lab.
- Python will be used for all programming in this course.
- Useful python related links:
  - Tutorials: https://docs.python.org/3/tutorial/
  - Language and API documentation: https://docs.python.org/3/

Course Objectives: The objectives of this course (paraphrased from the IDEA within course context) include:

- IDEA Objective 1: Gaining knowledge of the terminology, methods, trends of data structures and algorithms.
- IDEA Objective 2: Learning the fundamental principles and theories of algorithm analysis, including asymptotic analysis, complexity classes.
- IDEA Objective 3: Learning to apply data structures and algorithms to solving real-world problems.
- IDEA Objective 4: Developing professionally relevant skills, such as programming in the Python language, parallel programming concepts, among others.
- IDEA Objective 11: Learning to analyze and critically evaluate alternative data structures and algorithms.

Grading:		
	Written Problem Sets (approximately 5)	20%
	Programming Assignments (approximately 5)	35%
	Exams (2)	45%

Grading Scale:

A: at least 90.00	A-: at least 89.00	B+: at least 88.00
B: at least 80.00	B-: at least 79.00	C+: at least 78.00
C: at least 70.00	C-: at least 69.00	D+: at least 68.00
D: at least 60.00	D-: at least 59.00	F: less than 59.00

I reserve the right to adjust the scale at the very end of the semester. Such adjustments are rare, but will only be in your favor; and are highly unlikely to occur at the D-/F boundary. Note the 2 decimal places in the chart above (i.e., I do not round to the nearest whole number): e.g., unless I adjust the grade scale, an 89.99 is an A-, etc.

Exams: The exams are not explicitly cumulative, although many concepts covered in the course do build on earlier topics. For each exam you are allowed one sheet of notes (can use both sides) on a piece of paper no larger than 8.5" by 11" (letter sized paper). No other resources are allowed during exams. No books allowed. No calculators allowed. No computers allowed.

Make-Up Exams: Make-up exams will not be given (i.e., missed exam = 0), with the following exceptions:

1. Medical excuse: Provide documentation the first class you return after the missed exam. I suggest providing the documentation to the Wellness Center who will then contact all of the instructors of your courses.
2. Other institutional excuses: Situations may arise related to Stockton that prevents you from being able to attend an exam. In most such cases, you should be aware of the conflict beforehand. Thus, I must be notified one week prior to the missed exam. Send me e-mail via Blackboard with the details of the planned absence, and provide documentation (e.g., memo from sports coach, from other faculty sponsoring a field trip, etc).

Programming Assignments: There will be approximately 5 programming assignments. The time allotted to each will generally be 2 weeks, but may vary depending upon the amount of work required. You will be required to work independently, unless otherwise specified for the assignment. Programming assignments will involve implementing data structures and algorithms covered within the course, and either applying them to a given problem, or in some cases performing an experimental comparison of alternative algorithms for a problem.

Written Problem Sets: Written problem sets will include problems, exercises, and review questions related to the course topics. The written homework exercises must be done individually. You must show your work for full credit. A simple answer without showing the work that lead to it will receive no credit in most cases.

Due Dates: Programming assignments will be due electronically via Blackboard by midnight on the dates due. Written problem sets will be due by the start of class, and can be submitted either on paper or electronically in Blackboard. Late assignments will be penalized as follows: (a) 25% off if late by no more than 24 hours, (b) 50% off if late by no more than 48 hours, (c) 75% off if late by no more than 72 hours, and (d) a grade of 0 if late by more than 72 hours. The first time an assignment of each type is late (within 72 hours), the late penalty will be waived—i.e., you can be late with one programming assignment (within 72 hours of deadline) AND you can be late with one written problem set (within 72 hours of deadline).

Academic Honesty: Please familiarize yourself with Stockton's policy on academic honesty. Each violation will be penalized by a 0 on the relevant assignment/exam/etc, plus a 10 point penalty on your overall course grade. For example, if you have one violation, you'll have a 0 on that assignment or exam plus 10 points off your overall average, but if you have two violations, you'll have grades of 0 on the two assignments/exams/etc and 20 points off your overall average. Examples of violations include, but are not limited to: (a) any form of cheating on an exam or assignment, (b) passing off the work of another as your own (including other students, former students, code found on the Internet written by someone else, etc), (c) assisting someone in violating the academic honesty policy, (d) asking someone to assist you in cheating or other academic honesty violations (even if they refuse to help you cheat), etc.

Incomplete Policy: In general, no grades of incomplete will be given. The only exception to this rule is an institutionally documented medical emergency that necessitates your complete absence from Stockton for at least two continuous semester weeks. Additionally, you must be caught up on all work up to the point where your medical emergency began and currently in the "C" range or better overall at the point where the emergency began.

Tentative Schedule: This schedule is subject to change. Changes will be announced via Blackboard (and in class). If tentative exam dates change, they will be announced at least one week prior.

Date	Text and Topic
September 6	Course overview and introduction to algorithms (Ch. 1)
11	Introduction to Python
13	Introduction to Python
18	Review of basic algorithm concepts (Ch. 2)
20	Growth of functions (Ch. 3)
25	Graph Representations and Elementary Graph Algorithms (Ch. 22)
27	Graph Representations and Elementary Graph Algorithms (Ch. 22)
October 2	Minimum Spanning Trees (Ch. 23)
4	Minimum Spanning Trees (Ch. 23)
9	Single-Source Shortest Paths (Ch. 24)
11	Single-Source Shortest Paths (Ch. 24)
16	All Pairs Shortest Paths (Ch. 25)
18	All Pairs Shortest Paths (Ch. 25)
23	**NO CLASS: Preceptorial Advising Day**
25	Wrap up shortest paths and/or Review for Exam
30	**EXAM 1**
November 1	Multi-threaded / multi-process programming in Python
6	Multi-threaded / multi-process programming in Python
8	Multi-threaded Algorithms (Ch. 27)
13	Multi-threaded Algorithms: multi-threaded mergesort (Ch. 27)
15	Multi-threaded Algorithms: multi-threaded matrix multiplication (Ch. 27)
20	String Matching Algorithms (Ch. 32)
22	**NO CLASS: Thanksgiving**
27	String Matching Algorithms (Ch. 32)
29	NP-Completeness (Ch. 34)
December 4	NP-Completeness (Ch. 34)
6	NP-Completeness (Ch. 34)
11	Wrap up NP-completeness and/or Review for exam
13	**NO CLASS (finals week)**
18	**EXAM 2: (Slightly) different time: 12:30-2:30**