CSCI 4104: Data Structures & Algorithms II
Spring 2020 Syllabus

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

Office Hours: Monday/Wednesday 11:00am-12:15pm
Available other times by appointment; drop-ins also welcome

Course Time and Location: MWF: 12:45pm-2:00pm, 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:
- CSCI 3103: Data Structures & Algorithms I (C or better),
- CSCI 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/CSCI 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).
- Your discrete math textbook (if you still have it).

Other Required Resources:
- Python 3 (current version is 3.8.1, but any version that starts with 3 is sufficient): https://www.python.org/
  - Also acceptable: **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 3 will be used for all programming in this course. [Note: Python 2 is still actively used, but has differences in syntax and different libraries than Python 3. Make sure you use Python 3. If you submit programming assignments that require Python 2, you will get a 0 for those assignments. Python 2 is now permanently frozen as of January 2020.]
- Useful python related links:
  - Tutorials: https://docs.python.org/3/tutorial/
  - Language and API documentation: https://docs.python.org/3/

Stockton Computer Science Student Learning Outcomes: This course supports the following CSCI Student Learning Outcomes:

- 1. An ability to analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
    - a. Students will analyze a complex computing problem.
    - b. Students will apply principles of computing and other relevant disciplines to identify solutions to a complex computing problem.
- 2. An ability to design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
    - a. Students will design a computing-based solution to meet a given set of computing requirements.
    - b. Students will implement a computing-based solution to meet a given set of computing requirements.
    - c. Students will evaluate a computing-based solution to meet a given set of computing requirements.
- 3. An ability to communicate effectively in a variety of professional contexts.
    - a. Students will write technical documentation of a computer-based system, process, component, or program.
- 6. An ability to apply computer science theory and software development fundamentals to produce computing-based solutions.
    - a. Students will apply computer science theory to produce computing-based solutions.
    - c. Students will evaluate the effects of alternative data representations and algorithms on the performance of computing-based solutions.

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

- IDEA Objective 1: Gaining knowledge of the terminology, methods, trends of data structures and algorithms, including the fundamental principles and theories of algorithm analysis, such as asymptotic analysis, and 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. If I adjust the scale, it is done using a semi-automated approach involving clustering (i.e., "automated" == a program I wrote suggests a new scale based on all of the grades of the class; "semi-" == if that program's output is crazy, I ignore it and leave the scale alone; and "clustering" == a statistical technique). I never simply add a constant number of points to everyone's overall course score.

Other Grade Related Info: If you are in the B.S. in Computer Science, then this course is a core required course and must be passed with at least a C to count toward your degree requirements. If you are in the old B.S. in CSIS (CS concentration), then this course is just a CS elective and must simply be passed (any passing grade).

Exams: The exams are not explicitly cumulative, although many concepts covered in the course build on earlier topics. For each exam you are allowed one page 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. There will be no Python syntax or programming on the exams. However, there may be problems on the exams involving reading and/or writing algorithms in pseudocode.

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 written documentation (e.g., memo from sports coach, from faculty sponsoring a field trip, etc).
3. Based on University policy (https://stockton.edu/policy-procedure/documents/procedures/2030.pdf), if you are to be absent for a religious holiday on the date of an exam, you must notify me of that planned absence during the first 10 business days of the semester.

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. 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. Programming assignments will be implemented in the Python 3 language. Any assignments submitted that require Python 2.7 will receive a grade of 0.

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.

Yes, I am aware that there are websites containing the textbook solutions. A few graduate students at a school I won't name worked through all of the problems and posted their solutions online. I have a copy of this, so I will know if you use them. Please see the academic honesty policy in the syllabus. Also, approx. 33% of their solutions are either totally or partially incorrect; and 33% are incomplete (e.g., an answer without work, etc).

Due Dates: Programming assignments are due in Blackboard by midnight on the dates due. Written problem sets are due by the start of class, and can be submitted either on paper or in Blackboard. Late assignments are 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 is waived—i.e., you can be late with one programming assignment (within 72 hours of deadline) AND one written problem set (within 72 hours of deadline).

Academic Honesty: Please familiarize yourself with Stockton's policy on academic honesty. Each violation is 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. Example 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 or problem solutions 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. [Yes, I encountered that last one once in a General Studies course.]

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.

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

Date	Text and Topic
January 22	Course overview; introduction to algorithms (Ch. 1)
24	Course overview; introduction to algorithms (Ch. 1)
27	Introduction to Python
29	Introduction to Python
31	Introduction to Python
February 3	Review of basic algorithm concepts (Ch. 2)
5	Review of basic algorithm concepts (Ch. 2) / Growth of functions (Ch. 3)
7	Growth of functions (Ch. 3)
10	Graph Representations and Elementary Graph Algorithms (Ch. 22)
12	Graph Representations and Elementary Graph Algorithms (Ch. 22)
14	Graph Representations and Elementary Graph Algorithms (Ch. 22)
17	Minimum Spanning Trees (Ch. 23)
19	Minimum Spanning Trees (Ch. 23)
21	Minimum Spanning Trees (Ch. 23)
24	Single-Source Shortest Paths (Ch. 24)
26	Single-Source Shortest Paths (Ch. 24)
28	Single-Source Shortest Paths (Ch. 24)
March 2	Slack and/or Review for Exam
4	**EXAM 1: Part 1**
6	**EXAM 1: Part 2**
9	All Pairs Shortest Paths (Ch. 25)
11	All Pairs Shortest Paths (Ch. 25)
13	All Pairs Shortest Paths (Ch. 25)
16	**NO CLASS: Spring Break Week**
18	**NO CLASS: Spring Break Week**
20	**NO CLASS: Spring Break Week**
23	Multi-threaded Algorithms (Ch. 27)
25	Multi-threaded Algorithms (Ch. 27)
27	Multi-threaded Algorithms (Ch. 27)
30	Multi-threaded / multi-process programming in Python
April 1	Multi-threaded / multi-process programming in Python
3	Multi-threaded / multi-process programming in Python
6	Multi-threaded mergesort; multi-threaded matrix multiplication (Ch. 27)
8	**NO CLASS: Preceptorial Advising Day**
10	Multi-threaded mergesort; multi-threaded matrix multiplication (Ch. 27)
13	String Matching Algorithms (Ch. 32)
15	String Matching Algorithms (Ch. 32)
17	String Matching Algorithms (Ch. 32)
20	NP-Completeness (Ch. 34)
22	NP-Completeness (Ch. 34)
24	NP-Completeness (Ch. 34)
27	NP-Completeness (Ch. 34)
29	NP-Completeness (Ch. 34)
May 1	Slack and/or Review for Exam
4	**NO CLASS (finals week)**
6	**EXAM 2: (Slightly) different time: 12:45-2:45**