Syllabus

CSci 1913 (Section 10, Spring 2019)

Introduction to Algorithms, Data Structures, and Program Development

(Last updated 01/22/2019)

Essential information:

Meeting Times:	Lecture (010): MWF 11:15 - 12:05pm Akerman Hall 319 (http://www1.umn.edu/twincities/maps/AkerH) Lab (011): Mon 1:25 - 3:20pm Keller Hall 1-260 (http://www1.umn.edu/twincities/maps/KHKH) Lab (012): Mon 3:35 - 5:30pm Keller Hall 1-260 (http://www1.umn.edu/twincities/maps/KHKH)
Course Website:	Announcements and information are provided through the class Canvas website. The Canvas website will be updated regularly and will be the primary method for announcing any important class information. Therefore, students should plan on checking this for updates on a regular basis.
Instructor:	Name: Daniel Kluver Email: kluve018@umn.edu (mailto:kluve018@umn.edu) Office: 389 Shepherd Labs
Teaching Assistants:	Head TA: Saurabh Jogalekar Other TAs: • Tom, Busch

- Manu Padma
- Ryan Peroutka

Office Hours: See Office hours page on Canvas

Mark Allen Weiss. *Data Structures and Problem Solving Using Java.* Fourth Edition. Addison-Wesley. Boston, Massachusetts. 2010. ISBN–10: 0–321–54140–5. ISBN–13: 978–0–321–54140–6.

Textbook: For the first part of class we will be looking into the Python programming language and the following textbook may be helpful (it is available free online)
Allen Downey. *Think Python: How to Think Like a Computer Scientist.* Green Tea Press. Needham, Massachusetts. June 2014. **openbookproject.net/thinkcs/python/english2e. (http://openbookproject.net/thinkcs/python/english2e)**

Other texts may occasionally be referenced. These will be linked from the course Canvas website.

Prerequisites

EE 1301, CSci 1103, or CSci 1113. (EE 1301 is only accepted for EE or Computer Engineering majors)

CSci 1913 assumes that you have a background in basic computer science concepts including programming and problem-solving techniques. CSci 1913 builds upon these concepts and techniques.

Course Overview:

This course is an advanced object-oriented programming course. While we will cover many topics, a key focus of this course will be the implementation of abstract data types (stacks, queues, linked lists, hash tables, and binary tree structures) using the Java programming language. Additional topics will include: basic search and sorting algorithms, the principles of algorithm analysis, and programming in the Python programming language.

This course will be structured around a small number of substantial programming projects, and substantial weekly lab assignments.

Upon successfully completing this class, students should be able to use abstract data types in solving a variety of problems, and be able to implement their solutions in an object oriented programming language. Specifically, students should be able to:

1. Identify which abstract data type and/or algorithm could be useful in representing or solving a given problem, and why.
2. Given a computational problem, develop a functioning object oriented program in Java, and document its functionality in terms of the given problem.
3. Analyze implementations of a variety of searching and sorting algorithms with respect to their complexity.
4. Apply a variety of strategies to the testing and debugging of programs.
5. Apply consistent documentation and program style standards that contribute to the readability and maintainability of software.
6. Modify or specialize abstract data types to adapt them to problems that are not amenable to straightforward use of the abstract data types.
7. Compare alternative implementations of data structures with respect to performance.
8. Develop applications requiring file I/O and use of libraries using the Python scripting language.

CSci 1913 is a 4 credit course, with 3 hours/week of lecture, and a 2 hour/week lab.

Laboratories:

Most weeks, there will be a two-hour computing lab. Students will work with a partner during each lab, and both partners will receive the same number of points for their work. With the lab instructor's permission, you may work alone instead, but we recommend working with a partner where possible. Students need not work with the same partner each time.

- Treat lab assignments somewhat like weekly homework: while they may be discussed in a general sense with anyone, code should only be worked on between lab partners.
- Lab assignments must be submitted through the course Canvas site. All submissions are final: after you turn in an assignment, you cannot turn it in again.
- Late labs may be turned in up to 24 hours late, but with a 10% penalty. If your lab is more than 24 hours late, then it will not be accepted. If you must turn in a lab assignment late without penalty (perhaps because of illness), then ask your lab TA for permission.
- While we do not recommend missing labs, we understand that it is sometimes unavoidable. If you miss a lab, you must: (1) Get the lab assignment from Canvas. (2) Do the lab assignment on your own. (3) Submit the lab assignment before it is due.
- Please make sure you arrive at lab on time. Coming late can be disruptive to others since most of the labs are fairly full, and since you will likely be working in pairs. If you arrive too late to the lab, you might not be allowed to participate since labs are collaborative, and joining after others have done substantial work is disruptive and unfair to them.

- You should plan on staying in the lab until you finish all the problems or until the end of the lab time.
- TAs are in the lab to answer questions, and to check your work as you complete it. Feel free to ask them questions if any of the lab instructions are unclear, or if you need help in doing the lab problems.

Projects:

There will be three programming projects: one in Python, and two in Java. Projects must be done individually, although you may discuss them with others in a general way.

- Unless otherwise specified Projects are to be done individually. More details about what is and is not allowed is discussed in the "Academic Conduct" file on the course website. Please read this file carefully, and let the instructor know if you have any questions on it.
- If you have any questions about the meaning of a homework problem, please ask the professor or any TA. If you find any confusing typos or other errors in the homework problems, please notify the professor as soon as possible.
- Homework assignments will be graded on correctness, completeness, and style. Correctness and completeness refer to how well the program works. Style includes good design, readability (indentations, descriptive names for variables and procedures, appropriate use of blank spaces, etc.), efficiency, and useful comments.
- Projects must be submitted through the course Canvas site. All submissions are final: after you turn in a project, you cannot turn it in again.
- Late projects may be turned in up to 24 hours late, but with a 10% penalty. If your project is more than 24 hours late, then it will not be accepted. If you must turn in a project late without penalty (perhaps because of illness), then ask your lab TA for permission.

Examinations: There will be two midterm examinations, and a final examination. All three must be done individually. They will be ‘‘open notes,’’ so you will be allowed to use anything represented on paper to help you answer questions, including class notes, lab assignments, program listings, and Xeroxed pages (this may not be a complete list). You will not be allowed to use your textbook, or any other books. You will also not be allowed to use electronic devices, including calculators, cell phones, computers, ebook readers, Google glasses, smart watches, and tablets (this may not be a complete list). If you have questions about what is (or is not) allowed, then please ask me.

Quizzes: There will be approximately 9 quizzes over the course of the semester. These are to help you practice thinking about the more conceptual parts of this class’s material (algorithm analysis, debugging, comparing and contrasting different data structures and implementations). These will be delivered through the course Canvas site. You will have one attempt at each quiz. Quizzes should be taken individually and are open notes.

Office Hours: Office hours are a key component of success. They permit flexible one-on-one interaction not possible in class. The instructor and TAs encourage you to visit. Note: you are welcome to go to *any* of the TA office hours. That is, you are not restricted to the office hours of the TAs in your lab section.

- Email is best for short, well-defined questions. For example, emailing a TA an entire program and

asking why it doesn't work is not an efficient use of email.

- The TAs' role is to help you become a good programmer (not to do your programming for you.) So when you come to office hours be prepared to help the TA help you. For example, if you are having trouble getting started on a problem, be prepared to tell the TA what you tried so far (e.g., do you have an outline of the solution? have you read the relevant part of the textbook or reviewed any relevant parts of lecture?), and why you think it isn't right or where specifically you are stuck. If you are having trouble debugging a program tell the TA what specific error you are getting, what part of the program you think is causing it, what about the error you don't understand, and what you have tried to do to find or fix the error.
- Office hours are usually congested on project due dates. So start on the projects early so you will have ample time to ask any questions you have.

Grading

Here is the amount each of the items will contribute to your overall grade:

Labs	10%
Quizzes	5%
Project 1	10%
Project 2	10%
Project 3	10%
Midterm 1 (Fri. March 1)	15%
Midterm 2 (Fri. April 12)	15%
Final Exam (Final Exam 1:30-3:30 p.m., Tuesday, May 14)	25%

Please note the important dates above carefully, as make-ups will be given only under extreme circumstances. If you know that you will miss an examination, then you must contact The Instructor well before that examination to schedule an alternate date. In general, the earlier you let the teaching staff know about a conflict, the more permissive and accommodating we can be. If you miss an examination without prior notice then you must submit a written statement (by email to Daniel Kluver) that adequately

without prior notice then you must submit a written statement (by email to Daniel Rudver) that adequately explains your absence before an alternate exam time will be scheduled. Explanations like ‘‘I forgot there was a test’’ or ‘‘I overslept’’ or ‘‘my car wouldn’t start’’ may not be accepted. Examinations on alternate dates may be different from those on the scheduled dates: they may be easier or harder.

Grading for this course is on an absolute scale, so that the performance of others in the class will not negatively affect your grade. Final grades will be assigned based on the following scale:

95.0% -- 100.0% A
90.0% -- 95.0% A-
85.0% -- 90.0% B+
80.0% -- 85.0% B
75.0% -- 80.0% B-
70.0% -- 75.0% C+
65.0% -- 70.0% C
60.0% -- 65.0% C-
55.0% -- 60.0% D+
50.0% -- 55.0% D
0% -- 50.0% F

For S/N grading, a satisfactory grade (S) requires a weighted score of 60 or above.

Incompletes: will be given only in very rare instances when an unforeseeable event causes a student who has completed all the coursework to date to be unable to complete a small portion of the work (typically the final assignment or exam). Incompletes will not be awarded for foreseeable events including a heavy course load or a poorer-than-expected performance. Verifiable documentations must be provided for the incomplete to be granted, and arrangements for the incomplete should be made as soon as such the unforeseeable event is apparent.

Expected effort and participation: This is a four credit class that involves considerable effort. Most students find that this class requires 10 -- 15 hours of work during most weeks.

Policies

Withdraws: You are free to withdraw from the class up to the end of the tenth week of classes. Withdrawing thereafter is up to the college, and is not automatic. If you are not doing as well as you had hoped and are considering withdrawing, please do so by that date.

Scholastic conduct: The amount of collaboration allowed on assignments will be explained in the assignment rules or in a separate file posted to the class Canvas page. In general, you are free to discuss assignments with others, but you must work out and write your own solutions. Copying others' answers, or letting another person copy your answers is a serious situation and can result in failing the course. Additional explanation of academic conduct is in the academic conduct file that is posted to the class website. If you have any questions about what is and is not allowable in this class, please ask the course instructor.

Disability Accommodations: We desire to make learning rewarding and fun for all students and make every attempt to accommodate anyone who has a desire to learn. If you require special classroom or test-taking accommodations, you need to contact the University Disability Services and also notify the instructor as soon as possible at the start of the semester (no later than 3 weeks prior to the first examination).

Students Mental Health and Stress Management: As a student you may experience a range of issues that can cause barriers to learning, such as strained relationships, increased anxiety, alcohol/drug problems, feeling down, difficulty concentrating and/or lack of motivation. These mental health concerns or stressful events may lead to diminished academic performance or reduce a student's ability to participate in daily activities. University of Minnesota services are available to assist you with addressing these and other concerns you may be experiencing. You can learn more about the broad range of confidential mental health services available on campus via http://www.mentalhealth.umn.edu/ **(http://www.mentalhealth.umn.edu/)** .

For More information:

For more information about this course please visit the class canvas page or email the course instructor Daniel Kluver at his email <kluve018@umn.edu **(mailto:kluve018@umn.edu)** >.