COMP 410 section 001

Data Structures and Analysis

Spring 2020

Course Web page **http://www.cs.unc.edu/~stotts/COMP410-s20/**

Course Instructor: **David Stotts**
http://www.cs.unc.edu/~stotts
office: 144 Brooks *phone:* (919) 590-6133
hours: Tues 1:00 to 2:30; or email for an appointment

Prerequisites: COMP 401
Meetings: Class meets MW, 1:25 to 2:40 in SC 103 (Stone Center building)
First class is Wednesday, Jan. 8, 2020.
Teaching Assistants: See course web site or Piazza for TA names, offices, and hours.

Texts: Data Structures and Algorithm Analysis in Java, M.A. Weiss, 3rd. edition, Addison Wesley, 2012; ISBN-13: **978-0132576277**
(2nd ed. will work fine if you can get one; used copies are around)

This text is also available to purchase as an eText here. It can be cheaper that paper.

Email: Direct all class emails to PIAZZA. **This means you must register yourself in the Piazza site for this class. There is a Piazza folder "questions for instructor" for posting questions to the instructor and TAs/LAs**

Do not email my personal UNC account

Do not use TA or LA personal email accounts; do not use old COMP 410 accounts you might find… no one is monitoring those.

Email to accounts will get lost in the huge pile of other stuff we receive every day. I will communicate with you through Piazza announcements and replies to your private Piazza posts. I will also communicate with you via your email addresses given in Connect Carolina faculty central for the class. Please get in the habit of scanning your email often for information from me about the class. The titles of the emails will contain "COMP 410".

Target Audience: This course is intended for a student who has had a software development oriented first course in programming. It is taken in the first or early second year of a major program. It provided more extensive practice in developing programs, solving problems, and in designing algorithms that are ***practical and efficient***. We program in Java so Java knowledge and experience is assumed.

Timetable: The course web site contains a calendar showing the order of topics, due dates for assignments (as we give them) and exams, and other dates to note (like holidays). This is an initial fairly accurate estimate, but it may change at the discretion of the instructor, as we adapt to specific class events and progress.

Software:

1) **Web site:** Most course material will be distributed only on the web at the course web page. Use it as a clearing house for all course information, including this syllabus.
2) **Sakai:** We will use Sakai for submitting class assignments and for returning grades.
3) **Java:** We will develop **Java** programs in this class. While we don't prevent you from using your favorite IDE, we do encourage use of Eclipse. Our programming assignment

test cases will be distributed for Eclipse, and the instructions for using these oracles are in Eclipse. Your work will need to be submitted as Eclipse projects.

4) **JavaScript/Bricks:** I will also be programming some in class using Bricks, my own JavaScript system (you do not need to know JavaScript to pass this class, and you will not be required to write JavaScript programs). You may bring your laptops to class so you can follow along with my development of our data structures and algorithms. Active learning is very effective.

5) **Poll Everywhere:** We will start or end many class meeting using Poll Everywhere. Please go to https://www.polleverywhere.com and register, if you have not already done so for another course. I will be asking review questions and other questions designed to spur class discussing. Part of your class grade will come from your participation in these polls. The class polls will be found here as I generate them: https://www.pollev.com/comp410

6) **Piazza:** I have set up a Piazza site for this course. Please go register yourself to use it. Please use the site for discussions of the work we are doing in class. Post questions, answer questions, it is a social resource that previous classes have found very useful. We are also using the "questions for instructor" folder in Piazza in place of class email.

Supplies: You should have some means of backing up your programs (e.g. cloud store, flash drive, SD chip, CD/DVD writer, etc.). ***Do not allow your dog to eat your homework.***

Course personnel

Include graduate teaching assistants (TAs), undergraduate learning assistants (LAs), and the instructor. They have the following duties:

- LA: LAs will hold regularly scheduled office hours and will assist in the grading of your programming assignments and exams. Office hours will be to assist you with your programming assignments or with any other understanding of class materials.
- TA: TAs will hold regularly scheduled office hours; they will be available by email as well as at their office. The graduate TAs will be reviewing the Piazza site as well and will assist with discussions there. Programming questions about algorithm design as well as any questions that cannot be handled by any attendants in campus computing labs should be taken to the TAs or LAs (or to the instructor). Problems with, or questions about, course details and administration should be initially discussed with a TA. Grading questions on assignments should be directed to the TAs before discussing it with the instructor.
- The instructor: The instructor is available for assistance with any academic or administrative problem that your TAs can't handle, or to discuss issues of academic performance or class problems. Grading issues with exams that are not satisfied by the TAs should be addressed to the instructor. Office hours are shown above.
- *PLEASE DO NOT CALL THE INSTRUCTOR or TAs or LAs AT HOME.*

Learning Objectives

This course will teach you how to organize the data used in computer programs so that manipulation of that data can be done efficiently on large problems and large data instances. Rather than learning to use the data structures found in the libraries of programming languages, you will be learning how those libraries are constructed, and why the items that are included in them are there (and why some are excluded).

- You will gain familiarity with important categories of problems that are commonly encountered in software development, and will learn what data organizations will allow practical and efficient solutions to those problems.
- You will learn the basics of how to describe and analyze the performance and efficiency of your algorithms; this will prepare you for the upper level course in Algorithms Analysis.
- You will demonstrate the concepts you learn by encoding them in correct Java programs.
- You will gain more proficiency in basic programming and in constructing larger programs than you have been doing in your intro classes.
- You will practice the induction proof method learned in your background computational math course.
- You will learn the basic structure of the Java Collections library and demonstrate an understanding of its utility by employing some of its basic components in your later Java programs for more advanced data structures.
- Mastery of the data structures taught in this course will prepare you to study higher level areas where those data structures are heavily used: operating systems, networking, graphics, image processing, compilers, databases, robotics to name a few.

Calendar and Deadlines

See the class website for the class calendar. It contains topics to be covered each class day and linked supporting materials (PPTs, readings, etc.). See the website also for due dates of each programming assignment, along with detailed specifications.

Requirements and Policies

Exams: There will be two written in-class examinations – one midterm (see online class calendar) and a final (Thursay April 30, at 12:00 noon, per the UNC exam calendar). The in-class exams will be closed book, closed notes, closed computers, closed cell phones, closed smart watches, closed Google glass, closed classmates, closed everything other than your brain. I will supply all paper for the exam. We cannot give make up (early) exams for student who wish to travel early. **UNC sets the exam day/time, it is known here at the beginning, and that is the day/time we all have to work with.**

Grading: Approximately 25% of your grade will be based on homework/programming assignments; approximately 35% on the midterm exam, and 38% final exam. 2% will be awarded based on participation and correctness in the Poll Everywhere review/discussion questions. *This is subject to revision*.

Poll Everywhere: As noted, 2% of your final grade is based on your replies to Poll Everywhere review questions in class. This credit will be assigned as follows. If you get 0% correct you get 0 points added. If you get 1% to 20% correct, you get 0.4 point added to your final grade. If you get 21 % to 40% correct, then you get 0.8 points added, and so forth. You will earn the full 2% is you get between 81% and 100% of the Poll Everywhere questions correct. This "slack" scale means we will not reopen questions for people who miss class. The slack in the method will allow you to miss a few questions without affecting your results.

Programs: There will be approximately 6 programming assignments (5 graded, and 1 ungraded). Each assignment will have a specific date/time deadline; late programs will be penalized as follows: 10% per class day, which means 20% a week. For example, if an assignment is due Monday at 5pm, then it will have 10% taken off for being late up until

Wednesday at 5pm (the next class day); then it will be another 10% off up until the following Monday at 5pm (the next class day). It is always better to turn in something, even if it is very late, than to turn in nothing.

Your programs will be graded for both correctness and style. You will turn in an electronic copy of your program (through Sakai usually). Correctness will be determined by the TAs and/or instructor actually running the program. The style grade is determined from viewing the source code. More information about style requirements will be given later in the rubric for each assignment. Each program will be graded on a 100 point, negative scale. That is, you start with 100 points and we deduct points for errors of various kinds. Programs assigned later in the semester may be weighted a bit more heavily than the earliest programs.

Honor Code and Joint Work: I am very serious about the honor code. It's easy to cheat in many COMP courses thanks to the digital world; it's also very easy for us to detect plagiarism. So don't do it! Observe the University Honor Code. You are encouraged to discuss your work together for better understanding of the course material and assignments***. But do your own actual coding by yourself***. Unless specifically instructed otherwise on an assignment, you may NOT collectively write a program with a friend or classmate and then two or more people submit copies (or near copies) of the same code. We have software that searches for code that is plagiarized. We use it.

Although working together can enhance your learning, too much reliance on others can be disastrous at exam time. You are free to use any code that was presented in class, in help sessions, or in the course texts without permission or citation. Do not cut and paste Java code from the web for solving the problems we work on. You may study such code, but write out for yourself any that you wish to adapt; cite your sources in a comment. This will *enhance* your learning. You may NOT use material from other student from previous offerings of this course.

Incompletes: Incompletes will be given only in dire emergencies such as illness or a family emergency. Documentation (such as a physician's note, dean's note, etc.) will generally be required. Falling behind in your work is not an emergency.

Class Attendance: You are responsible for material we discuss in class. Attendance will assist you in learning the material. Attendance will also allow you to participate in Poll Everywhere, worth 2% of your final grade. This 2% could well push you over a line into a higher grade category so class attendance will assist you. Given the size of this class, we cannot consider each individual case and adjust grades that are close to a line at the end. So please use the Poll Everywhere opportunity to earn an easy 2%.

Students who make a habit of skipping classes often consume large amounts of TA time trying to get the information that the instructor provides in class. TA time is a limited critical resource; if you seek assistance from the TA on assignments and have not been attending classes you may be asked to wait while students who have attended are assisted first.

TA assistance: Our TA/LA staff will be happy to assist you in their office hours with your work. Please come prepared to ask questions to receive assistance. This means being with you the programming that you have done. Do not come to office hours to begin your work. Bring work with you so you can ask meaningful questions about your work and your progress. Start your assignments early. Do not wait until a couple days before they are due – you may not be able to get assistance if you wait.

Grading Criteria for Programs

- **Correctness.** Programs will be expected to be 100% functionally correct. This will require you to think through all the possible types of input and then test for correct handling of each type. We will give you an "oracle" for each assignment that will contain some test cases for the code being developed. The primary purpose of the oracle is to give you assurance that your code will go through our grading software without breaking something. The oracle is NOT a comprehensive list of test cases. It is NOT necessarily the case that if your code passes our oracle then your code is correct. Our grader will contain many more test cases than the ones given in the oracle. We expect you to THINK about your code and its behavior under ALL possible inputs, and to write code AND TESTS CASES OF YOUR OWN to make sure the code is behaving as it should.
- **Clarity and style**. Programs must be readable by humans. This requires that they be based on clear thought and be well-presented and well organized. A good presentation requires well-written code, careful formatting and good comments. Choose descriptive variable and class names. Choose good algorithms (the main purpose of this class!)
- **Adherence to assignment parameters**. There will be specific details in most assignments, such as "use a recursive algorithm" or "write your code to this interface". Pay attention to these instructions if you wish full credit. Being functionally correct does not imply full credit.

We use the following letter grade scale based on percentage:

```
[93-100] A
[90-93)  A-
[88-90)  B+
[83-88)  B
[80-83)  B-
[78-80)  C+
[73-78)  C
[70-73)  C-
[68-70)  D+
[60-68)  D
[00-60)  F
```

Disclaimer

The instructor reserves the right to make changes to the syllabus as presented here, including assignment and exam dates. Any such changes will be announced as far in advance as possible.