Data Structures + Algorithms

Greenwich Academy

2020-2021

Instructor(s):	Dr. Monica E. Ortiz	**Block:**	B
Email:	mortiz@greenwichacademy.org	**Classroom:**	Ramsing 2
Office:	US 112		

Welcome! This course is designed to take your knowledge of computer science and confidence in coding to a new level. In a university, this course would typically be the second course in the introductory series (a course similar to AP CS A being the first). One focus will be DATA: how to contain it, how to process it, and how to use it to make conclusions about interesting things. A second focus will be ALGORITHMS: how to analyze their efficiencies and the considerations that go into choosing the right tools for the job. As with most things coding, you can use multiple languages to learn these topics. We will use Java.

Course Outline:

1. Data + Using Files
2. Java Collections
3. Arrays and ArrayLists
4. Lists, Sets, and Maps
5. Recursive Algorithms, a second look
6. Algorithm Complexity
7. Stacks and Queues
8. Trees
9. Classic Algorithms

Course Resources:

1. **Class Canvas page:** Be sure you have your Notification settings sorted out. This is where announcements, assignments, grades, etc. will be posted. For any assignments requiring you to produce code, this is where you will upload said code.
2. **Additional Resources:**
   - Building Java Programs by Reges and Stepp This is a great resource that has taught me a lot.
   - Thinking in Java by Bruce Eckel *(Free online)*

When we can meet: My free blocks this year are C, D, and G (less reliably). Additionally, I am generally available after school. The best way to find a time is to talk with me before or after class or to email me ahead of time.

Grading: Each assignment will be given a point value, which will be averaged into a grade for Homework and for Quizzes/Tests/Projects. Problem Sets will count for 40% and Quizzes/Tests/Projects will count for 60% of the term average.

Grades are yours. As such, grades will be posted on Canvas so there should never be any uncertainty about your grade. If you feel there has been a mistake, please come find me immediately (i.e., *not* the day before the term ends) so that I can remedy the situation!

Late assignments: they happen. I get it. You get one freebie (you may turn in one assignment late by 1 day without penalty) per semester. Otherwise, a point deduction of 15% /day will apply. If there are extenuating circumstances, let me know and I will work with you. However, you typically will have 1 week or more to complete assignments, so this will also be taken into consideration.

Collaborative Work Policy and Getting Help: Part of learning to programming and, indeed, of becoming a computer scientist, engineer, developer, etc. is working with your peers. I encourage you to work together in some capacity. This will allow you to learn topics in a different way than I have taught during class, will better prepare you for working with others in the real world, demonstrates alternative ways to program the same thing, and allows you to get practice at reading and interpreting each others' code. *These are all good things.*

However, there are also disadvantages. One such disadvantage is that you may be tempted to forego the necessary steps of learning in order to get your work to "done". For example, asking a friend "What is wrong with my code?" could result in a fruitful discussion between two classmates in which the teaching student learns how to better interpret another's code and the student being taught is helped to better debug his or her code. It may also result in the student teacher saying, "Here, just take a look at mine." *Borrowing/copying code is not permitted* as it does not help either party learn and is plagiarism. Please be sure to keep your discussions with other students to how to do something, not to adaptations of others' code.

When having difficulty with a particular concept, problem, or project, I encourage you to go through the following steps:

1. Re-read the problem in case you missed something.
2. Use the debugging methods we learn in class:
    - Add print statements to output different variable values.
    - Bypass specific variables to better control the conditions.
    - Set breakpoints and use the built-in debugging tools in BlueJ to step through your code line-by-line.
3. Decode any compile-time errors that are generated.
4. Look line-by-line for any syntax errors (e.g., missing semicolons, unmatched brackets, capitalization errors, etc.).
5. Ask me, or a fellow student, to help you go through these steps.

Academic Honesty: Universities have long established policies for dealing with breaches of academic ethics standards, and you will be held to a similar standard while in this class. **Under no circumstances are you permitted to present anyone else's work as your own. Period. Hard stop.** I understand that it can be frustrating to have trouble arriving to the right answer, but there is more than enough support available to help any student who is having difficulty. Penalties for academic dishonesty may include, but are not limited to, grade reductions and referrals to school administration. While there may be reasons to cheat, there are no *good* reasons to cheat. The penalty to your grade and to your soul for cheating is far more severe than turning in the wrong answer.

One contributing factor to academic dishonesty is falling behind on your work, i.e., procrastination. If you fall too far behind, it becomes impossible to catch up without cheating. I would much rather you come talk to me about the situation instead. First, I will respect your honesty. Second, I may be willing to make an accommodation to help you get back on track, like pushing back a due date for some agreed-upon grade penalty. Of course, if you have some legitimate issues preventing you from getting your work done, I will work out an arrangement with you provided that you do the responsible thing — find an adult you trust to help you, be it me, another teacher, your advisor, or a school administrator.