

# CS310: Data Structures (Spring 2020)

## Section 006

### 1. Course Basics

#### 1.1 Prerequisite:

C or better in CS211 and MATH 113.

#### 1.2 Professor Information:

Professor	Email	Office
Dr. Maya Larson	larson@gmu.edu	ENGR 2706

#### 1.3 Textbook:

- Mark Allen Weiss, **Data Structures & Problem Solving Using Java**, 4th ed., Addison-Wesley, 2010. [Errata](#)
- It is assumed that students read the scheduled sections prior to each lecture.

#### 1.4 Software & Programming Requirements

- You will need a computer for this class with some modern operating system capable of running a Java compiler. It is recommended that you bring a laptop computer to class as there will be some class room programming activities. If you do not bring a laptop computer to class, these activities will be done in a paper version. Lab computers are also available on campus, see campus resources below.
- The prerequisite for this course is C or better in CS 211. Students need to have previously developed a significant degree of skill in programming (program organization, coding, documenting, testing and debugging). Students should be previously acquainted with basic complexity analysis ("big O") and able to build abstract data types using Java classes.

## 1.5 Discussion board: Piazza

Course schedule, announcements, lecture slides, assignments, discussion. GTA/UTA contacts and office hours will be on Piazza too. Sign up yourself by following this link:

<https://piazza.com/class/k5okg41byce6ha?cid=5>

- **The discussion board on Piazza is required reading for all projects.** You MUST read the discussion board for clarifications and project updates.
- All instructor and TAs can view all material on Piazza.
- No sharing answers or code solutions to assignments on the discussion board. You can include code in private posts to the instructors (visible to all professors, GTAs, and UTAs). See Honor code section below for more details.
- Email course staff only for logistical issues such as meeting outside of office hours, missing lab/lecture, grading disputes, medical situations, etc. Email addresses are listed on Piazza.
- Do not e-mail course staff about programming problems; use the discussion board. *Project help questions sent via email are of extremely low priority, as they were sent to the wrong place.*

## 1.6 Blackboard

- Grades will be posted to Blackboard.
- Projects assignments will be submitted on Blackboard.

## 1.7 Office hours

Office hours are listed for all staff on the CS website

<https://cs.gmu.edu/academics/>

- Students must make significant, individual effort on all projects before coming to see a GTA/professor. Waiting until the last minute, in the expectations that the entire project will be explained during office hours, is not feasible.
- Office hours are often crowded - do not rely on them for last minute help, as we cannot guarantee that we will be able to spend significant time with every student. You should use the discussion board on piazza for posting questions.

## 2. Course Description

### 2.1 Course topics:

CS 310 continues the study of data structures from CS 211. Students will learn how to approach larger and more challenging programming problems than the projects in CS 211. Programming is a significant part of this course and students should expect to spend a good deal of time on the programming projects. The course also introduces a variety of data structures and illustrates the types of problems for which they are useful.

Tentative topics to be covered include:

- Lists, Stacks, and Queues
- Hash Tables and Hashing
- Trees and Basic Tree Algorithms
- Graphs and Basic Graph Algorithms

### 2.2 Course Outcomes:

In CS310, students will:

- Gain an ability to analyze the runtime efficiency of algorithms that create and manipulate data structures.
- Gain an ability to compare data structure trade-offs such that the student is able to choose appropriate implementations for a given data structure and appropriate data structures for a given application.
- Reinforce their knowledge of basic data structures (linked lists, dynamic array lists, stacks, and queues) and extend their knowledge to more sophisticated data structures, including hash tables, balanced trees, priority queues, and basic graphs.
- Implement and use learned data structures to solve real computing problems. This includes substantial reinforcement of language-specific techniques for abstract data types and design and debugging techniques.

# 3. Grading

## 3.1 Assessment

Category	Percent
Programming Assignments	40%
Participation	5%
Quizzes	5%
Midterm Exam	25%
Final Exam	25%

There will be no make-up or extra-credit assignments at the end of the semester; your grade should be a measure of your semester-long progress.

## 3.2 Programming assignments:

Students will receive a number of programming projects during the semester. These programming projects will be a primary focus of your grade - each one should take multiple sessions of coding, with questions asked in between. Don't be surprised if you're spending 20+ hours on each one. All project grades will be averaged together evenly. Programming projects are **INDIVIDUAL** work.

- **Submissions**

- All projects are to be submitted to Blackboard. You can submit your work an unlimited number of times to Blackboard, only the last version will be graded. It is a good idea to submit often to back up files and provide a record of your work.
- Turning in the wrong files will likely result in a zero. It is each student's responsibility to double check their work by downloading submitted files and verifying that they are correct.
- **Broken Code == Bad Scores.** Any code turned in that does not compile, specifically on Java 1.8 is likely to receive a zero. If the grader is able to quickly fix your code, you might get some points back. If the grader cannot immediately spot and fix the issue, you'll be fortunate to get any points at all.

- **Deadlines**

- Projects can be turned in at most 24 hours late, no exceptions.
- Submitting a project after the deadline will incur a 5% penalty. One minute late is the same as 23 hours and 59 minutes late.
- Blackboard being unavailable is not an excuse for turning in a late assignment; in the rare situation that the website is somehow unavailable or giving the student an error, the student **MUST** email their submission to both the GTA and the professor before the deadline, otherwise it will be considered late.
- **Back up your work.** Use a backup service such as DropBox (or any cloud service), email files to yourself, submit files on Blackboard as you work. Don't be the student who forgot to (frequently) back up your work!
- Catastrophic computer failure will not be cause for an extension. Every semester multiple students' computers die, are stolen, or otherwise 'lose' projects. Back up your work.

- **Honor code**

- Programming projects are considered **individual** efforts, therefore no sharing of code and/or discussion of problem solutions are allowed with anyone except the TAs or the professor. Student projects will be manually and automatically assessed for cheating.
- You may not use any Internet resources to create code or algorithms, besides the textbooks, the slides, and Piazza, unless otherwise specified. However, you are free to look up the syntax errors you encounter online, to gain an understanding of what the syntax error means. The projects we're doing this semester can be directly solved using techniques discussed in class, and no outside material is needed unless otherwise noted.
- **It is your responsibility** to lock your computers with a password, to not post your code to websites like Pastebin that are publicly accessible, to guard your USB drives and computers, to not upload your files to someone else's computer, etc. You will be liable for any access gained to your code.

### **3.3 Participation:**

- 5% of the semester total points is based on participation activities.

### 3.4 Exams:

- Exams are closed book/notes unless specified otherwise by instructor. They will be entirely paper and pencil - no computers.
- All students must have their GMU identification available on testing days.
- The final exam is cumulative.
- Early or Late/Makeup Exams require exceptional circumstances (as judged by the instructors) and must be arranged in advance. Vacation plans are not exceptional circumstances.
- If you miss an exam due to a university-accepted excused absence (such as an illness or car accident the day and time of the exam), you must notify your professor within 24 hours of your absence to make arrangements for a makeup, and bring approved documentation with you when you take the make-up exam. Failure to follow either of these policies will result in a zero. We may also elect to allow the final exam to count the extra amount to give you a sort of do-over.

### 3.5 Contested Grades:

- If you feel points have been incorrectly deducted, contact the grader. For projects, that is your GTA. For exams, that is your professor.
- Contesting of grades on any/all submissions must be requested within one week of receiving the grade (on Blackboard). No grade changes will be considered subsequent to that deadline.

## 4. Honor Code

- All students are expected to abide by the [GMU Honor Code](#). This policy is rigorously enforced. Cheating will be prosecuted and result in a notification of the Honor Committee as outlined in the GMU Honor Code. **Sharing, collaboration, or looking at any code or algorithm for the programming projects that is not your own, including resources from Internet, is considered cheating.**
- The computer science department has an [CS Honor Code Policies](#) that you are subject to particularly for our course. The document helps you to understand how GMU honor code applies to programming and CS, what constitutes cheating in the CS setting. It clarifies some scenarios that are unique to our sorts of assignments. Make sure you read and familiarize yourself with these rules.

- We take the honor code seriously. Any attempts at copying or sharing code, algorithms, or other violations of the honor code will be reported to the Office of Academic Integrity. We use automated software to flag suspicious cases, and then review them to find the cases that must be submitted to the Office of Academic Integrity. The penalty for cheating will always be far worse than a zero grade, to ensure it's not worth taking the chance. **Confirmed cases of cheating almost always translate into course failure.**

## 5. Learning Disabilities

Students with a learning disability or other condition (documented with GMU's [Office of Disability Services](#)) that may impact academic performance should speak with the professor ASAP to discuss appropriate accommodations. Even if you don't know whether you plan on utilizing the accommodations for any assignment/test, it's in your best interest to prepare and get documented ahead of time.

## 6. Campus Resources

- [Computer Labs](#) – there are several freely available computer labs on campus
- [GMU Learning Services](#)
- [Office of Disability Services](#)
- [Counseling and Support Services](#)
- [English as a Second Language \(ESL\) Writing Support](#)