

## *Data Structures and Algorithms*



Trinity College  
Library  
Univ. of Dublin

## William D McQuain

Email: `wmcquain@cs.vt.edu`  
Zoom: `virginiatech.zoom.us/my/wmcquain`  
Office: 634 McBryde Hall (N/A for Spring 2021)  
Zoom Hours: see course website

### The Plan

- lectures will be recorded and posted on Canvas
- all other course materials will be posted on the main course website
- all of my office hours will be held on Zoom
- some office hours will be *en masse*, like a help session
- some office hours will be one-at-a-time
- tests will be offered via Canvas, using the Lockdown Browser and Respondus Monitor
- I will NOT be available for in-person consultations

<http://courses.cs.vt.edu/cs3114/Spring21/>

<https://canvas.vt.edu/courses/125285>

## CS 3114 Data Structures and Algorithms

Advanced data structures and analysis of data structure and algorithm performance. Sorting, searching, hashing, and advanced tree structures and algorithms. File system organization and access methods.

Course projects require advanced problem-solving, design, and implementation skills.

Having successfully completed this course, the student will be able to:

- Choose the data structures that effectively model the information in a problem.
- Judge efficiency trade-offs among alternative data structure implementations or combinations.
- Apply algorithm analysis techniques to evaluate the performance of an algorithm and to compare data structures.
- Implement and know when to apply standard algorithms for searching and sorting.
- Recognize and apply design patterns, and make judgments about when a particular pattern will improve a design.
- Design, implement, test, and debug programs using a variety of data structures including buffer pools, hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and B-trees.
- Select appropriate methods for organizing data files and implement file-based data structures.
- Apply object-oriented design principles to data structures in medium-scale software systems.
- Apply design guidelines to evaluate alternative software designs.

Final grades will be based on the average achieved over the following :



Item	Weight	Dates
Programming Projects	36%	See website
Homework	24%	See website
Tests	10% each	Tentatively Feb 22 and Apr 12
Final Exam	20%	TBD

Instead of a midterm test, the course will involve a number of mini-tests, offered on selected Thursdays, via Canvas with the Lockdown Browser and Respondus Monitor.

Each mini-test will cover a small selection of course topics, and will take 40 minutes.

The final exam will be comprehensive, on a date to be determined, via Canvas with the Lockdown Browser and Respondus Monitor.

## Lectures

PowerPoint course notes will be posted on the CS-hosted website (not Canvas).

I will make one (or more) lecture recordings for each set of notes.

You should review each set of PowerPoint notes before viewing the related lecture recording(s).

The lecture recordings will be posted on the course Canvas site.

## Assignments

You will submit all of your assignment solutions via the Curator system.

Non-programming assignments will be graded manually by the course TAs, and feedback will be provided via the Curator.

Programming assignments will be evaluated by using stand-alone automated grading packages, which will be supplied to you.

## Office Hours

I will not be holding any “in-person” office hours.

Instead, I will host my office hours on Zoom; my plan is that:

- some office hours will be restricted to a particular course (I'm also teaching 2505)
- some office hours will use the Zoom waiting room; during those I will usually interact with one student at a time
- some office hours may be mass meetings
- see the CS 3114 Forum board for announcements

## Help Sessions

I will also host occasional help sessions on Zoom:

- each of these will be devoted to a particular assignment, and will be "open" mass meetings
- I hope to incorporate some Q/A interactions in these sessions
- details will be announced on the CS 3114 Forum board

## Administering Tests

Obviously, the two tests and the final examination will be given online.

I anticipate that each test:

- will be hosted on Canvas
- will require your use of the Lockdown Browser with Respondus Monitor

Instructions for this will be supplied in advance of each test.



**CS 2114 Software Design and Data Structures**

**CS 2505 Computer Organization I**

**Math 2534 Discrete Mathematics**

**or**

**Math 3034 Introduction to Proofs**

**All students** must have completed each of the CS prerequisites with a grade of C or higher (C- is not acceptable).

There will be absolutely **NO** exceptions to these requirements.

Austin Chennault	<code>achennault</code>
Hemayet Ahmed Chowdhury	<code>hemayetahmedc</code>
Archi Dasgupta	<code>archidg</code>
Mohammed Elarnaoty	<code>marnaoty</code>
Brannon King	<code>brannonking</code>
Rifat Sabbir Mansur	<code>rifatsm</code>

All TA office hours will be held on Zoom.

The course website includes a link to a Google Calendar showing those hours.

Office hour changes will be reflected on that calendar and announced on the course Forum Board.

Cecilia Duan	dcecilia19
Joseph Dyer	josephdyer
Luke Jordan	bruce1i
Lalitha Kuppa	klalitha
Kara Probasco	karap17
Danny Sharp	dannys4

All TA office hours will be held on Zoom.

The course website includes a link to a Google Calendar showing those hours.

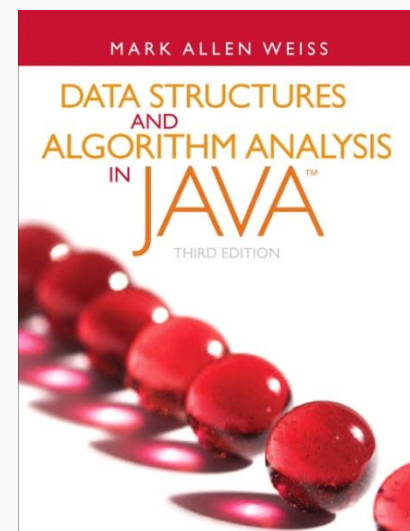
Office hour changes will be reflected on that calendar and announced on the course Forum Board.

## Required:

*Data Structures and Algorithm Analysis in Java, 3rd Edition*

Mark Allen Weiss, Addison-Wesley, ©2012

ISBN 978-0-13-257627-7



## Recommended:

*CS 3114 Course Notes, Spring 2021 Edition* available at:

[courses.cs.vt.edu/cs3114/Spring21/](https://courses.cs.vt.edu/cs3114/Spring21/)

Lecture and other recordings available on Canvas:

[canvas.vt.edu/courses/125285](https://canvas.vt.edu/courses/125285)

## Grade Scale

The usual 10-point scale will apply (subject to any curve). A final average of 90% will guarantee an A-, 80% will guarantee a B-, and so forth.

## Curve

A grade curve may or may not be employed in this course. The application of a curve is dependent upon class performance on tests, projects and homework. The decision to utilize a curve rests entirely with the course instructor.

## Statute of Limitations

Any questions about the grading of an assignment must be raised with your instructor within two weeks after the graded assignment has been made available to you.

The purpose of this is to ensure that assignments are graded correctly, and in a timely matter, not for you to make corrections to your submissions.

## Late Penalties for Projects

Project solutions can be turned in after the posted deadline, in which case a per diem penalty will be assessed.

The penalty will be 10% per day.

## Extensions

Any request for an extension must be made, preferably by email, at least 24 hours prior to the due date. Reasons must be valid (generally events beyond your control) and documented.

Late submissions will not be given any credit if submitted after graded assignments or solutions have been released.

## Homework

Late submissions of homework assignments will only be allowed with an extension.

## File Formats

Each assignment in this course will include precise instructions for the type of file that is to be submitted (e.g., zip, tar, text, jar, etc).

Such requirements may stem from a variety of concerns, none of which are negotiable.

If you do not understand how to create a file of the specified type, discuss that with a member of the course staff.

## Submission Contents

Each assignment in this course will include precise instructions for exactly what must be submitted to us.

If you make a submission that is incomplete, or otherwise incorrect, we will grade that.

It is your responsibility to make sure that you submit the correct information.

It is our responsibility to grade what you submit, not to compensate for incorrect submissions.

## Special Treatment

There will be none.

There are many of you, and few of us, and I will not permit the TAs to spend extra time to compensate for your failure to follow the instructions that are given in the assignments.

## Evaluation of Correctness, Design and Implementation, and Documentation

The correctness of operation of your programming projects will be evaluated by executing your solution with test data constructed by the course staff. While some test data will be provided, there is no guarantee that data will cover all cases.

Each project will have certain explicit requirements for design and implementation. Your solution will also be evaluated for adherence to those requirements.

You are expected to enter CS 3114 with considerable understanding of good software engineering practice, and you are expected to apply those lessons here. That means that you may be penalized for failing to make good decisions, even if there are no explicit guidelines in the specification of a project.

You are also expected to incorporate professional internal documentation into your projects. See the Programming Standards page on the course website for some suggestions and samples.



## Test Environments

- All programming assignments submitted are required to compile with `javac` version 8 or later.
- All code testing will be performed on CentOS 8, as installed on the rlogin cluster. There should be no difference between that and Windows, since execution is actually taking place via the JVM.
- Unless specified otherwise, programs will only be tested under that environment.
- It is the **YOUR** responsibility to ensure that **YOUR** programs execute correctly in the appropriate environment; programs that do not will receive substantial deductions.



Students developing with a different Java major version are advised that I have tested my solutions with both Java 8 and Java 11 (my default), and noted no differences.

Your programs **WILL** be tested with the environment listed above. If it fails to compile, or exhibits incorrect behavior, we don't care that it may compile elsewhere, or appear to run correctly elsewhere.



## Why use the command line?

- All programming assignments submitted are required to compile with `javac` version 8 or later.
- The Eclipse IDE uses its own internal Java compiler; that may differ from the "real" Java compiler from Oracle.
- We will compile from the command line with the "real" Java compiler when we test your projects.
- It is the **YOUR** responsibility to ensure that **YOUR** programs execute correctly in the appropriate environment; programs that do not will receive substantial deductions.

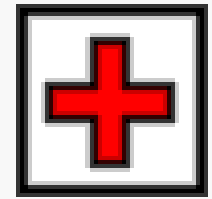


## How can I use the command line?

- You have considerable experience with the Linux command line from CS 2505.
- The basic principles are the same for the Windows command line.
- We will post notes on installing the JDK from Oracle and using it from the Windows command line. Things would be very similar on Linux.

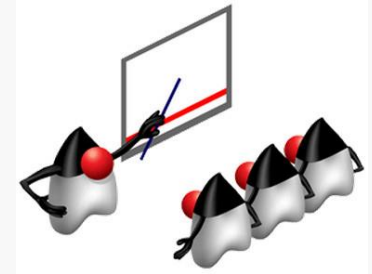
## General Issues

- CS 3114 classmates
- CS 3114 Forum board
- CS 3114 TA
- CS 3114 Instructor



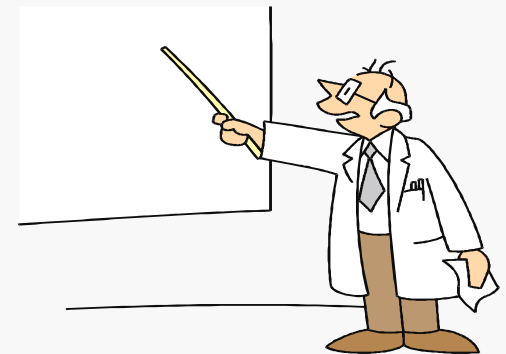
## Java Language Help

- CS 3114 Forum
- texts from earlier courses



## Lecture Instruction

Lectures will consist of presentations, applications, problems and solutions interspersed with classroom discussion.



## Backups

**Students are responsible for making backup copies of all their work in this (and all) courses.**

Loss of work due to hard drive failure is **NOT** an acceptable excuse. Backup copies of files on the same hard drive are not backup copies. Backup copies of files on second hard drives are also risky. Backup copies should be maintained on two separate distinct storage mediums, (e.g., hard drives and Zip disks).

Backup copies should be maintained until after the end of the term and students have received their course grade. (The Army lives by triplicate for a reason.)

Remember: Computer systems are mechanical devices.

Systems fail. Drives die. Bad sectors appear.

Network connections break.

Plan for it. It is inevitable!



The Undergraduate Honor Code pledge that each member of the university community agrees to abide by states:

“As a Hokie, I will conduct myself with honor and integrity at all times. I will not lie, cheat, or steal, nor will I accept the actions of those who do.”

Students enrolled in this course are responsible for abiding by the Honor Code.

A student who has doubts about how the Honor Code applies to any assignment is responsible for obtaining specific guidance from the course instructor before submitting the assignment for evaluation.

Ignorance of the rules does not exclude any member of the University community from the requirements and expectations of the Honor Code.

For additional information about the Honor Code, please visit:

<https://www.honorsystem.vt.edu/>

Commission of any of the following acts shall constitute academic misconduct. This listing is not, however, exclusive of other acts that may reasonably be said to constitute academic misconduct. Clarification is provided for each definition with some examples of prohibited behaviors in the Undergraduate Honor Code Manual located at <https://www.honorsystem.vt.edu/>

### CHEATING

Cheating includes the intentional use of unauthorized materials, information, notes, study aids or other devices or materials in any academic exercise, or attempts thereof.

### PLAGIARISM

Plagiarism includes the copying of the language, structure, programming, computer code, ideas, and/or thoughts of another and passing off the same as one's own original work, or attempts thereof.

### FALSIFICATION

Falsification includes the statement of any untruth, either verbally or in writing, with respect to any element of one's academic work, or attempts thereof.

### FABRICATION

Fabrication includes making up data and results, and recording or reporting them, or submitting fabricated documents, or attempts thereof.

## MULTIPLE SUBMISSION

Multiple submission involves the submission for credit—without authorization of the instructor receiving the work—of substantial portions of any work (including oral reports) previously submitted for credit at any academic institution, or attempts thereof.

## COMPLICITY

Complicity includes intentionally helping another to engage in an act of academic misconduct, or attempts thereof.

## VIOLATION OF UNIVERSITY, COLLEGE, DEPARTMENTAL, PROGRAM, COURSE, OR FACULTY RULES

The violation of any University, College, Departmental, Program, Course, or Faculty Rules relating to academic matters that may lead to an unfair academic advantage by the student violating the rule(s).

An exhaustive list of Honor Code violations would be impossible to present here, but among other things, each of the following is a flagrant violation of the Virginia Tech Honor Code, and violations will be dealt with severely (Honor Court):

- Working with another student to derive a common program or solution to a problem. **Unless explicitly stated otherwise, there are no group assignments in this course.**
- Discussing the details required to solve an assignment. You may not share solutions, or collaborate in the creation of a solution.
- Copying source code (programs) in whole or in part from someone else.
- Copying files from another student's disk or lab account even though they might be unprotected.
- Editing (computer generated) output to achieve apparently correct results.

It is acceptable to discuss an assignment with classmates in a general way, i.e., to discuss the nature of the assignment. In other words, you may discuss with your classmates what your solution is required to accomplish but not how to achieve that goal using C, MIPS32 assembly, or other relevant tools. In no way should the individual statements of a program or the steps leading to the solution of the problem be discussed with or shown to anyone except those people cited in the following statement.



Feel free to discuss the homework assignments and your program source code with the teaching assistants assigned to your section of CS 3114, and the instructor. The discussion of your program source code must be limited to these people. Note that this specifically excludes discussions of your program source code with other students (even if they are not enrolled in CS 3114), or with tutors. Privately hired tutors are not an exception to this requirement, nor are athletic or other tutors provided by the University.

Copies of all submitted work are retained indefinitely by the Department. Submitted programs are subjected to automated analysis for detection of cheating.

If you have any question as to how the Honor Code applies to this class, remember that:

- Any work done in this class must be done on an individual basis.
- Credit will be given only for work done entirely on an individual basis.
- Do not make any assumptions as to who can provide help on a programming assignment.
- All submitted work is archived. All submitted programs will be subjected to automated cheating analysis via the MOSS system.

Evidence indicating the violation of the policies stated above will be submitted to the Honor Court.

It is much easier to explain a poor grade to parents or a potential employer than to explain an Honor Court conviction.

If you have questions or are unclear about what constitutes academic misconduct on an assignment, please speak with me.

I take the Honor Code very seriously in this course. The normal sanction I will recommend for a violation of the Honor Code is an **F\*** sanction as your final course grade. The F represents failure in the course. The “\*” is intended to identify a student who has failed to uphold the values of academic integrity at Virginia Tech.

A student who receives a sanction of **F\*** as their final course grade shall have it documented on their transcript with the notation “FAILURE DUE TO ACADEMIC HONOR CODE VIOLATION.”

You would be required to complete an education program administered by the Honor System in order to have the “\*” and notation “FAILURE DUE TO ACADEMIC HONOR CODE VIOLATION” removed from your transcript.

The “F” however would be permanently on your transcript.

In recent terms we have observed a new behavior regarding cheating.

Some of the projects we use each semester may have been used in previous offerings of the course.

A small number of students have submitted solutions that were based (in whole or in part) on solutions submitted by other students in previous offerings. Be advised:

- That is cheating.
- When we do the cheating analysis for a project, we include submissions from previous offerings as well as the current term. (We save everything.)
- If we detect this form of cheating, the students from the previous terms will also be charged with violating the Honor Code.
- A charged student will not receive a degree until the charges are resolved.
- The University does make provisions for cases in which a charged student has already graduated. You do not want to discover how this works, so safeguard your code for the long term!

At some point you will very likely want to make samples of the code you write available to potential employers. There is absolutely nothing wrong with that!

But if you do that, you need to be sure you're not making it easy for other students to access your code.

GitHub is an interesting service, but you need to use the protection options that the free student accounts offer.

And there are alternatives...

Students are encouraged to address any special needs or special accommodations with me during the first two weeks of the semester, or as soon as you become aware of your needs.

Those seeking accommodations based on disabilities should obtain a Faculty Letter from the Services for Students with Disabilities office (540-231-0858) located in Lavery Hall, Suite 310) <http://www.ssd.vt.edu/>.

If you need adaptations or accommodations because of a disability (learning disability, attention deficit disorder, psychological, or physical), if you have emergency medical information to share with the instructor, or if you need special arrangements in case the building must be evacuated, please meet with the instructor as soon as possible.