CS 3350 Fall 2017 – Data Structures and Algorithms

Contact Information

Instructor	Dr. Robert Dyer
Office Hours	Tues/Thurs 1–4pm OR by appointment
E-mail	rdyer@bgsu.edu
Office	HAYES 244
Phone	(419) 372–3469

Prerequisites

Math 2220/3220 and grade of C or better in CS 2020

Class Meeting Time

Section 1001: Mondays/Wednesdays/Fridays, 11:30am-12:20pm, HAYES 117

Section 1002: Mondays/Wednesdays/Fridays, 12:30-1:20pm, HAYES 117

Textbook

The textbook **MUST BE** purchased through Perusall.

======> DO NOT PURCHASE A BOOK <======

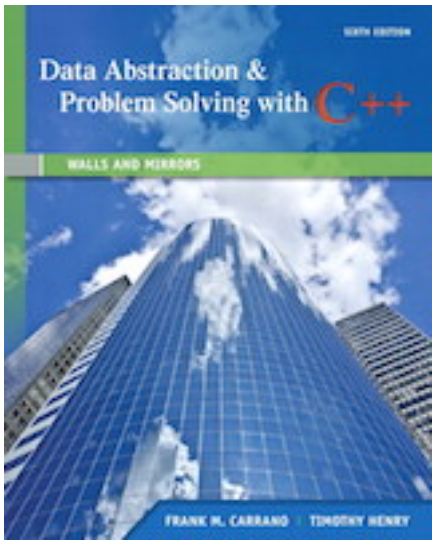

Figure 1: Book

======> DO NOT PURCHASE A BOOK <======

"*Data Abstraction & Problem Solving with C++: Walls and Mirrors,*" 6th Edition (2013), Carrano and Henry, Pearson, E-book.

The textbook **MUST BE** purchased through Perusall.

======> DO NOT PURCHASE A BOOK <======

Outcomes for the course

After successfully completing CS 3350, students will be able to say:

- I can solve computation problems using recursion.
- I can implement and apply stacks, queues, trees and other custom data structures.
- I can create generic functions and classes.
- I can understand algorithmic complexity (e.g., Big-Oh notation)
- I can understand the relationship between data structures and algorithms.
- I can understand the design tradeoffs (e.g., code complexity and performance) in data structures and algorithms.

Grading

The final grade will be composed of the following weights. (The instructor reserves the right to make changes at any time.)

Assessments

Item	Points Each	Total
Exams *(3)*	55	165
Final Exam	85	85
Labs *(10)*	25	250
Programming Assignments *(5)*	50	250
Perusall Annotations *(20)*	5	100
Readiness Assessments *(20)*	5	100
In-class Activities *(10)*	5	50
Total		**1000**

Grading Scale

Point Range	Percentage	Grade
920 - 1000	92 - 100%	A
820 - 919	82 - 91%	B
720 - 819	72 - 81%	C
620 - 719	62 - 71%	D
0 - 619	below 62%	F

Teams

This course relies heavily on team work. First day of class we will form teams for the entirety of the semester. You will work with your team for in-class activities and the RAAs. Team members are expected to all contribute equally.

The reading assignments are also collaborative in nature, as every student will be put (randomly, and different for every assignment) into a group of approximately 15 students who can view and respond to your annotations.

Labs are mostly done individually, but are collaborative in nature. I encourage students to help each other during labs and students are free to work in pairs. **Each student must submit their own solution.**

Finally, all programming assignments are done in pairs. The first lab will be an exercise to become accustomed to pair programming. All students will work in pairs and **each student in a pair will submit the same solution**. There will be periodic evaluation of each person's individual performance to ensure fairness.

Assessments

Readings

Readings are absolutely essential to learning in this course. Almost every lecture there are readings due, meaning there are approximately 2 readings due each week. Readings will be done on the website Perusall (note: **only access Perusall via the links in Canvas!**) which let's you annotate the text being read. These annotations are required and graded. The system will automatically grade each annotation. It keeps the 5 highest annotations. Thus, to ensure the best possible score you should aim for 7-10 good annotations.

Annotations could be you summarizing a piece of text for the class, asking an insightful question, answering other student's questions, finding problems with the text, etc. Your annotations should be spread out through the whole assigned reading (and not all in one small area). You are graded on 3 criteria: quantity of annotations, average quality of annotations, and spacing of the annotations in the text.

Readiness Assurance Assessments (RAAs)

To help me gauge where the class is at, we will have in-class assessments for each reading. These contain a small number (5) of questions that anyone who completed their reading assignment should be able to answer. Students will take the assessment first by themselves. You will then collaborate with your team members and re-take the same assessment as a team. The score will be the average of your individual score and the team score.

In-class Activities

Instead of providing traditional lectures, the majority of class time will be spent doing individual and/or small group activities. The goal is to give students hands-on experience with the material while I and other students are available to help clarify concepts students may be struggling with. Although these activities are graded, the grade is based on attempting to complete the problem and not necessarily on solving it. It is vital that students show up for class having read the assigned material and ready to work!

Labs

There are several programming labs, completed in class on Wednesdays in the Hayes 020 lab. Labs may not be completed outside of your assigned lab time.

Programming Assignments

There will be several programming assignments completed outside of class and individually. To receive credit for your assignments, they must be submitted online by the due time. **There are no late submissions allowed.** Partial credit will be given for any completed portion of the assignment, so be sure to submit on time even if you are not finished with the assigment!

Exams

There are a total of 3 exams throughout the semester and a final exam during finals week. Exams will consist of a variety of question types, including multiple choice, true/false, short answer, short programming questions, and interpreting code. The final exam is approximately 60% new material and 40% cumulative questions. All exams will be completed on a computer in Hayes 020.

Technology

Canvas

The syllabus, all assignments, and due dates are posted on Canvas. Your grades will also be available on Canvas throughout the semester. Canvas is the main entry point for this course - everything you need to do is linked and organized from the Canvas course. Always start there!

Plickers

Each student will be assigned their own Plickers card. Plickers cards are 3D barcodes, and depending on the orientation of the card (4 possible sides can face up) you are able to respond to questions with answers A, B, C, or D. This allows quick, interactive feedback from the class. I also use these to quickly record attendance near the start of each class.

Perusall

The reading annotation website we use is Perusall. To access Perusall, please click the link for a particular reading assignment in Canvas. This should open the reading (in Perusall) in a new tab/window. THERE IS NO NEED TO CREATE A USER ON PERUSALL (and doing so will actually break your grading!).

Perusall provides some documentation to explain what kind of comments we are looking for.

Mimir (section 1001 only!)

Section 1001 will use the Mimir platform, an online platform for compiling and running our C++ programming assignments. It can be accessed from a link in Canvas. Mimir will allow you to submit your solution to the problem, compile it on their server, and then run against pre-defined tests and show if your solution passes those tests or not. You will have the ability to re-submit to try for a higher grade. You can also edit your code directly in their platform using their online IDE.

BGUnix (section 1002 only!)

Section 1002 will use BGUnix, a mainframe computer on campus, to type in, debug and run our C++ programming assignments. It can be accessed from any lab computer on campus or from a personal/home computer using PuTTY with Windows (PC) or Terminal on a Mac.

Other documentation about BGUnix (getting a BGUnix account, using editors on BGUnix, etc.) are available at https://www.bgsu.edu/its/services/class/bg-unix.html

Course Policies

Withdrawal Deadline

Friday, November 11, 2016. University policy states that after this date, anybody withdrawing from the course will have the grade automatically turn into an F.

Office Hours and Help

Please check Canvas, Canvas inbox, and your BGSU email regularly. [You may have your Canvas messages forwarded to your BGSU/other email, and have your BGSU email forwarded to another favorite email address, if necessary, but do check it (multiple times) daily.] I forward my own Canvas messages to my BGSU email and check my BGSU email multiple times everyday (with rare exceptions). I will do my best to accommodate you ASAP, even if outside my posted office hours and without appointment. In general, if you need to see me in my office outside of my regular office hours, please make an appointment to ensure I am available.

Attendance

Students are expected to attend each class and be on time. I take attendance at the start of each lecture. I typically use good attendance as a factor when considering final grades. I reserve the right to penalize students up to 1% of their final grade, per absence, for more than 3 un-excused absences.

Make-up policy

If you cannot take an exam as scheduled, you (or an authorized person, only in case you are unable to do so) must contact me ahead of time with the reason. Make-ups are considered typically for health emergencies only.

Academic honesty

All coursework (unless done in pairs/groups) for this class is expected to be YOUR OWN work. The MINIMUM penalty for copying someone's work (including current classmates, students from a previous offering of the course, or postings found on the web) or knowingly allowing someone to copy your work is a zero for the homework/project/exam/paper/presentation. The offense is also reported to the dean of your college. Plagiarism detection tools will be used in this course. I will follow the Department's policies and the University's code of academic conduct as defined in the BGSU Student Handbook. For details refer to:

1. Department of Computer Science Academic Honesty Policy
2. BGSU Code of Academic Conduct
3. The Academic Charter, section B-I.G

Disability Policy

In accordance with the University policy, students with disabilities must verify their eligibility through the Office of Disability Services, 38 College Park Office Building, 419–372–8495 (http://www.bgsu.edu/disability-services.html). Contact me as soon as possible this semester to arrange any accommodations needed to assist with your success in this course.

Religious Holidays

It is the policy of the University to make every reasonable effort allowing students to observe their religious holidays without academic penalty. In such cases, it is the obligation of the student to provide the instructor with reasonable notice of the dates of religious holidays on which he or she will be absent. Absence from classes or examinations for religious reasons does not relieve the student of responsibility for completing required work missed. Following the necessary notification, the student should consult with the instructor to determine what appropriate alternative opportunity will be provided, allowing the student to fully complete his or her academic responsibilities (The Academic Charter, section B–I.F–4.b).

Tentative Course Schedule

Week	Day	Date	Topics
1	M	Aug 21	Introduction
	W	Aug 23	Ch. 1 Data Abstraction
	F	Aug 25	Interlude 1 C++ classes
2	M	Aug 28	Interlude 2 Pointers
	W	Aug 30	Lab 1 - *Hayes 020*
	F	Sep 1	Ch. 2 Recursion
3	~~M~~	~~Sep 4~~	Labor Day - No class
	W	Sep 6	Lab 2 - *Hayes 020*
	F	Sep 8	Ch. 3 Array-based Implementation
4	M	Sep 11	Ch. 4 Link-based Implementations
	W	Sep 13	Lab 3 - *Hayes 020*
	F	Sep 15	Interlude 3 Exceptions
5	M	Sep 18	Exam review
	W	Sep 20	**Exam 1** - *Hayes 020*
	F	Sep 22	Ch. 6 Stacks
6	M	Sep 25	Ch. 7 Implementing Stacks
	W	Sep 27	Lab 4 - *Hayes 020*
	F	Sep 29	Ch. 8 Lists
7	M	Oct 2	Ch. 9 List Implementations
	W	Oct 4	Lab 5 - *Hayes 020*
	F	Oct 6	Ch. 10 Algorithm Efficiency
8	~~M~~	~~Oct 9~~	Fall Break - No class
	W	Oct 11	Lab 6 - *Hayes 020*
	F	Oct 13	Ch. 11 Sorting Algorithms
9	M	Oct 16	Exam review
	W	Oct 18	**Exam 2** - *Hayes 020*
	F	Oct 20	Ch. 12 Sorted Lists
10	M	Oct 23	Ch. 13 Queues/Priority Queues
	W	Oct 25	Lab 7 - *Hayes 020*
	F	Oct 27	Ch. 14 Implementing Queues
11	M	Oct 30	Ch. 15 Trees
	W	Nov 1	Lab 8 - *Hayes 020*
	F	Nov 3	Ch. 16 Tree Implementations
12	M	Nov 6	Ch. 17 Heaps
	W	Nov 8	Lab 9 - *Hayes 020*
	~~F~~	~~Nov 10~~	Veteran's Day - No class
13	M	Nov 13	Exam review
	W	Nov 15	**Exam 3** - *Hayes 020*
	F	Nov 17	Interlude 6 Iterators
14	M	Nov 20	Interlude 7 STL
	~~W~~	~~Nov 22~~	Thanksgiving Break - No class
	~~F~~	~~Nov 24~~	Thanksgiving Break - No class
15	M	Nov 27	Ch. 18 Dictionaries
	W	Nov 29	Lab 10 - *Hayes 020*
	F	Dec 1	Ch. 19 Balanced Search Trees
16	M	Dec 4	Ch. 20 Graphs
	W	Dec 6	Lab 11 - *Hayes 020*
	F	Dec 8	Exam review
17	M	Dec 11	1:15pm-3:15pm **Final exam** - *Hayes 020* (*section 1002*)
	T	Dec 12	10:45am-12:45pm **Final exam** - *Hayes 020* (*section 1001*)