

SYLLABUS
CS 163: Data Structures
Summer 2020

Prerequisite: **CS162 or consent of instructor**
This means you should have experience writing complete programs in C++ and be proficient creating classes and coding linear linked lists

Instructor:	Karla Steinbrugge Fant		
E-mail:	karlaf@pdx.edu		
Office hours:	T/TH 2:00-3:00	Zoom ID:	257 779 553
Private			
SLACK Channel:	https://psu-ccu9286.slack.com		
Live Streaming			
Sessions:	T/TH 3:30-5pm	Zoom ID:	965 856 56062 password: cs163
Homework:	Monday 2-3:50	Zoom ID:	503 725 5394 password: hwrec
Recitation	Tuesday 2-5:50	(Homework Rec starts June 26th)	
	Wednesday 2-3:50		
	Thursday 2-5:50		
	Friday 12-3:50		

Text:	Data Abstraction and Problem Solving with C++, Carrano (6 th or 7 th edition)
Canvas:	All required lecture videos, powerpoint slides, sample code, and lab materials may be found on Canvas under “Modules” Link: pdx.instructure.com
Disabilities:	If you have a disability and are in need of academic accommodations, please notify the instructor immediately to arrange needed support. This includes any accommodations required for taking examinations. All DRC exams must be taken at the same time as the class exams except when otherwise pre-authorized. Such pre-authorization should take place at least 24 hours prior to the class exam.
System & Compiler:	CS Linux (linux.cs.pdx.edu). C++ language implemented by the g++ compiler. GNU GCC C++ compiler (g++) in the default -ansi mode Use the C++ standard 98 guidelines (-ansi). Editors MUST be either: vi, vim, emacs <i>No IDEs are allowed when working in C++;</i> <i>Not allowed: Dev Cpp, Visual Studio or xcode</i>

Prior Knowledge expected:

CS163 is designed for students who have already have programmed in C++ previously. In this class we use C++ to learn about data structures applying the syntax previously learned. You should already be able to program using C++:

- (a) classes, pointers and dynamic memory (new, delete)
- (b) functions with pass by reference
- (c) linear linked lists (traversal, creation, removal)
- (d) and, understand recursion at a conceptual level

Course Description and Goals:

Data abstraction with formal specification. Elementary algorithm analysis. Basic concepts of data and its representation inside a computer. Linear, linked, and orthogonal lists; tree structures. Data structures are implemented as data abstractions. Sorting and searching strategies. Data management.

To acquaint students with structures used in C++ for the storage and manipulation of data. The concept of data abstraction and the problem of building implementations of abstract data types are emphasized. Both static and dynamic implementations of major structures are presented and the advantages and disadvantages of each are discussed. Structures include lists of several types, stacks, queues, trees, binary trees, B-trees and graphs. Recursion and key transformation (hashing) are examined. Students are encouraged to examine algorithms and to make judgments about the practical and social application of these algorithm concepts to large scale programming projects; the course stresses the importance of quantitative methods.

Remote for Summer 2020

This summer, CS163 is offered as a remote class. There is a mix of required content videos and live streaming sessions. The content videos must be watched prior to the live streaming sessions. The live streaming sessions will take place during our scheduled class period (T/TH 3:30pm) and typically end by 5pm. All live streaming and lab sessions will take place via Zoom.

Attendance to the live streaming sessions is part of your success in this class; missing sessions means that success will be jeopardized. Material will be covered in the live streaming sessions that cannot be found in any other format. Consider the live streaming sessions as required. If you are unable to attend these sessions, contact karlaf@pdx.edu to arrange an alternative (such as through watching recordings of these sessions). Approval for such an arrangement will be made on a case by case basis and in such situations the student agrees to watch all recordings for the missed live streaming sessions.

Students understand that live streaming sessions may be recorded remotely and this is being done with your permission. Such recordings are limited for use for just students in this course and may not be distributed to others outside our class roster. **It is important to note that the lab sessions are not recorded.**

Online examinations (including proficiency demonstrations) will take place using a recorded proctoring service. Students are required to have a webcam (USB or internal), a microphone, and internet access for these events. It is recommended to preplan to ensure that the space is private or semi-private and that the recordings that will be made do not invade any third-party privacy rights.

All students must review the syllabus and requirements including online terms and testing requirements. Enrollment in this course is an agreement to abide and accept all terms.

Communication in a Remote Environment

One of the pieces we miss by being remote is building a community. We can use the live streaming sessions to “get to know” each other. This can help by having your webcam on whenever you find it comfortable to do so and interacting with other students in the chat session. Of course, when using chat we are missing body language cues and immediate feedback; therefore, it is very important to be aware of online etiquette. This ensures that the message you intend to convey is received correctly.

When writing messages in the chat, keep in mind these points:

- **Be respectful.** It is essential to keep in mind the feelings and opinions of others, even if they differ from your own.
- **Discuss material related to the class.** Remember chat sessions are not the same as personal social media.
- **Be aware of strong language, all caps, and exclamation points.**
- **Be clear with humor and sarcasm.** Is what you are saying clear?
- **Remember we are all in this together. We all have real world struggles as we navigate this extraordinary time.**

Proficiency Demonstrations:

- Twice a term we will be evaluating coding proficiency using live programming examinations called Proficiency Demonstrations.
- These are performed by appointment
- Every student in CS163 must show proficiency in programming in C++ and data structures, using linux with either vi, vim, or emacs
- Proficiency demos must be passed to receiving a passing grade in CS163
- Proficiency demos are scored as:
 - E (exceeds, passing),
 - P (proficient, passing),
 - PW (pass with warning, passing)
 - IP (in-progress, non-passing), ** maybe retaken once at midterm time
 - U (unsatisfactory, non-passing)
- A PW (pass with warning) at the final time is an indicator that students should continue to practice before taking CS202. We recommend enrolling in a data structures practice lab first (CS299)
- There are no re-tests available for the final proficiency demo.

Lab Sessions:

The lab sessions are where we reinforce the materials learned in lecture. It is where concepts will be practiced prior to applying them to the larger individual programming assignments. *It is expected that all students will attend the lab section enrolled in, each week.*

Labs begin week #1. Attendance is required. (July 3rd is a holiday and PSU is closed)

Starting the third week, completion of a prelab quiz on Canvas is required in order to attend lab. Prelabs help ensure that students are ready for the lab experience; if you can't complete the prelab then it means additional studying should be done prior to the lab so that you can make the most of your time with our technical assistants. Additional help can be obtained through homework recitation, office hours or by appointment.

Some key points about labs include:

1. **Attendance** is required and labs are scored pass/no pass.
2. There are six lab sessions this term. Refer to the Course Plan to learn which lab is being performed each week.
3. The lab materials can be found on Canvas in the Lab shell (CS163L) and it is organized by the Lab number.
4. **Read** the background for each lab, **prior** to attending the corresponding lab!
5. **REQUIRED: Perform the prelab quiz** on Canvas prior to attending the lab
6. **Plan** to fill out all of the questions in the lab materials as you progress through the lab.
7. **On your own, make sure** to perform and practice all of the vim exercises assigned

Individual Programming Assignments:

There are three components to your programming assignments this term. We have design writeups with test plans, the programming assignment code, and an efficiency writeup which is turn in with the code.

1. Design writeups and Test Plans

- a. There are 3 written design write-ups that are turned in **prior** to the first three programs. Included with each design write-up must be a test plan that outlines how you intend to test the software. A test plan "form" is available on Canvas. The design write-ups must be a minimum of 600 words.
- b. **Late write-ups** will be accepted by the late date **with a 5% deduction**.

2. Individual Coding:

- a. There are 5 programming assignments
- b. All programs must be created individually and written in C++ on linux.cecs.pdx.edu
- c. **Every** program must be completed and submitted.
- d. **Passing scores** are expected **on average** to pass CS163.
- e. Turning in work by the late deadline is a 5% deduction
- f. **Assignments** may **not** be turned in later than the late due date
- g. **Programs incorrectly** uploaded will receive a 5% penalty.

3. **Efficiency Writeup:**

- a. Every program must have a minimum 600 word write-up explaining the efficiency of the code that was written. This is 20% of the program's grade.
- b. These are turned in with your program
- c. Please do not combine the writeups with your tarball or zip file. They should be uploaded separately to Canvas.

4. Follow the STYLE SHEET!

- a. Program style and comments is 20% of each program's grade.
- b. Each file must have your name and header comments describing the purpose of the code within the file.
- c. The file header comments should be a **paragraph**, at a minimum
- d. Each function must have header comments describing the purpose of the function and arguments. No exceptions!
- e. Make sure that each function that has a non-void return type returns a value through each possible path
- f. Always use the returned value when calling a function
- g. Avoid single character variable names, except for loop control variables and array indices
- h. Avoid while(1) or while(true)
- i. Avoid the use of break or return from within a loop
- j. Very few functions in CS163 should have void return types.

5. Each program will have **multiple files** (.h and .cpp files for C++) and have their name and header comments in each file.

- Group related functions and classes within a single file.
- Never implement functions in a .h file and never #include a .cpp file!

6. **Partial credit will be given for incomplete programs.** This means that it is better to turn in something, even if it doesn't work. If you find you are continually having problems meeting the due dates, make an appointment or visit office hours.

7. **Every assignment (programs and write-ups) must be submitted** to pass CS163.

- a. All code and written material must be your own work and may not be copied from the web or other students. Be careful to not plagiarize. Doing so will result in a zero on an assignment and a failure in the class. Receiving "too much help" is not a valid reason to receive a passing score.
- b. Never let anyone write the code for you! Never just copy code from the internet!

Overview of Grading Policies:

Demonstrate Proficiency in C++ <ul style="list-style-type: none">- Midterm Demo- Final Demo	Pass/No Pass	Must pass both demos with E, P, or PW
Lab Participation <ul style="list-style-type: none">- Prelab Quizzes Performed- Lab Code Submitted	Pass/No Pass	Attendance is required
Individual Assignments <ul style="list-style-type: none">- Design and Test Plan- Programming Projects	20%	All Programs must be submitted & have scores of 40% or above; the average score of all must be 65% to pass
Midterm Exams <ul style="list-style-type: none">- Midterm #1- Midterm #2	20% 20%	Each midterm score must be at least 65% or greater to pass
Comprehensive Final Exam	40%	Final exam score must be 65% or greater to pass the class

Grading Policies:

1. Two midterms, each worth 20% of your grade
2. The Final Exam is 40% of your grade

• Exams will all be closed book, closed notes.

- If a DRC exam is being taken, please email your teacher with a reminder that an exam is needed at the testing center. Do not expect an exam to automatically be sent without such email.
- For C or better in this class, you must receiving a PASS on all of the Pass/No pass components of this class (see the chart above)
- All 5 programs must receive a 40% or better individually and on average of 65% or better to pass.
- The midterms and final must receive passing scores to pass the class.
- Failure to turn assignments on-time or within the allowed late period will result in a zero for that assignment. Discuss extraordinary situations with your teacher.
- GRADING will be done near 90% (A-, A), 80% (B-, B, B+), 65% (C). Exact break points for grades will depend upon the overall class results. A No Pass on the proficiency demos or a failure to turn in an assignment will result in a non-passing grade (F, D-, D, D+).
- No Basis for a Grade applies when a student has not turned in any work and have not taken any exams. If you have complications and cannot finish the class, make sure to drop or withdraw. Otherwise you will get a grade in the class.
- INCOMPLETES will be given only when a minimal amount of work remains to be completed, only for a valid reason and only for a fixed time period. An Incomplete only applies in situations where the work already submitted has passing scores. Do not expect an incomplete in this class.

CHEATING:

Each student is expected to submit only original work. **Any person who violates these requirements will receive a grade of zero for an assignment which based on the above grade requirements will result in an F for the course.** A letter will be sent to the head of the CS Department.

The work you submit must be your own. It is not acceptable to hand in assignments in which substantial amounts of the material was done by someone else. You must be especially careful that in the process of discussing problems with other students that they do not inadvertently end up using your work. In such an event, all students involved will receive a zero on that assignment.

Students will receive a zero on an assignment if any of these activities take place:

1. Student provides proficiency demo questions to other students
2. Student provides proficiency demo solutions to other students
3. Student solicits (asks for) proficiency demo questions and/or solutions from other students
4. Student copies lab code from another student
5. Student accepts an assignment and/or program from another student
6. Student supplies an assignment and/or program to another student
7. Student posts the assignment and/or program on the web, social networking site, or Canvas or Slack discussions
8. Student shares their password with another student at PSU giving that student access to their assignments and/or programs
9. Students work together on assignments or turn in the similar assignments.
10. Student turns in work that was obtained from other sources such as the web, friends, tutors or technical assistants
11. Student leaves work available for others to copy from
12. Student attempts to purchase programs from others (in person or electronically).

Seeking Assistance

Be careful when seeking help from others. We recommend seeking help from (a) instructor, (b) TA's and lab assistants, and (c) Tutors. Use caution otherwise. Do not to share your code with others! Never accept code that was not written by someone else!

This means, NEVER accept code from someone else, even if it is a tutor!

- Never post your code in the Canvas or Slack discussions, the web, or social networking sites.
- Never give your assignments to other PSU students, regardless of their situation.
- Never email your code to anyone except your instructor.
- I recommend NEVER having someone else use your keyboard or type code for you when asking for help. Ask them to teach you what you need to know rather than do it for you!!!

Let's have a GREAT SUMMER!