SYLLABUS
CS 163: Data Structures
Spring 2018

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

Instructor:	**Karla Steinbrugge Fant**
E-mail:	**karlaf@pdx.edu**
Message Phone:	**503-725-5394**
Office Location:	**FAB 120-19**
Office hours:	**Tuesday and Thursday 2:30-3:20 in FAB 120-19**

Text: Data Abstraction and Problem Solving with C++, Carrano (6th or 7th edition)
D2L: Lecture notes and course power point slides are on D2L

Lab Materials: **Required.** Purchase from the PSU Book Store
The Linux and Vim Manual published for **2017-2018 academic year** and
The CS163 Lab Manual published for **Spring 2018**
(earlier editions are not acceptable)

PSU ID: Bring your PSU ID card to all lectures and labs; it will be used for attendance purposes.

Handouts:

All handouts, due dates, and assignments can be retrieved from D2L

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 quizzes and exams must be taken at the same time as the in-class quizzes and exams except when otherwise pre-authorized. Such pre-authorization should take place at least 24 hours prior to the in-class quiz or 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
No IDEs *are allowed when working in C++;*
Not allowed: *Dev Cpp, Visual Studio or xcode*

Expected: **Use of gdb with each C++ assignment**

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

Students should be proficient programming linear linked lists (traversal, insertion, and removal algorithms).

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.

Overview of Grading Policies:

Demonstrate Proficiency in C++ - Midterm Demo - Final Demo	Pass/No Pass	At PSU or by Proctor (must pass both demos with E, P, or PW)
Lab Participation - Prelabs - Lab Code Submitted	Pass/No Pass	Pre-labs and Lab code (Attendance to all but 1 lab required)
Graded Lab Manuals - Students will complete a grading rubric form	5%	Lab Manuals are graded twice in regard to completeness, relevance, and readability
Individual Assignments - Design and Test Plan - Programming Projects	25%	(All Programs (design + programs) must be submitted & have scores of 40% or above; the average score of all must be 65% to pass)
Quizzes and Midterm Exam - 2 Quizzes - Midterm Exam	5% 25%	At PSU or by Proctor (Midterm score must be 65% or greater to pass)
Comprehensive Final Exam	40%	At PSU or by Proctor **Final exam score must be 65% or greater to pass the class**

Mandatory Attendance

Students are required to attend all class meetings. A student is marked as having attended a class if the student ID has been scanned in the first 5 minutes of the class meeting. Students may miss two class meetings without penalty. After that, each missed class will result in one point being deducted from their final average in the class.

Important Class Expectations:

1. **Attending lectures is required to pass the class.**
    a. **PSU ID's** are scanned each lecture (at the **beginning** or **end** of the lecture)
    b. Arrive within the first 5 minutes to count as attending.
    c. It is expected that students will stay for the entire class period, until excused.

2. **Attending lab sessions is required to pass the class**
    a. **PSU ID's** are scanned at the **beginning** and **end** of labs.
    b. Arrive within the first 5 minutes to count as attending.
    c. Be prepared to stay the entire lab period.
    d. One lab session may be missed without making it up
    e. It is expected that students will program, write algorithms, and design test plans for the entire duration of the lab. If lab work is completed prior to the end of the lab period, practice questions may be assigned by the lab assistants.
    f. It is encouraged that students continue working on lab materials individually

3. **If you miss more than one lab session, makeups must be within a 1-2 week period.**
    a. Attendance to a makeup lab must be scheduled through https://cs163_makeup_lab.youcanbook.me/
    b. Makeup labs begin the third week of the term
    c. To makeup **more than two** labs, contact **karlaf@pdx.edu** to seek authorization.
    d. To makeup a lab beyond the 1-2 week time period, contact **karlaf@pdx.edu** to discuss alternatives. *Do not assume that an early lab can be made up late in the term!*

4. **Pre-lab exercises** must be brought to the labs already completed.
    a. The prelabs are required as part of your attendance.

5. **Lab Manuals** will be collected and graded twice during the term.
    a. It is expected that students will fill-out the lab questions in the manual as they progress through a lab.
    b. The lab manuals are graded for completeness and readability. Answers count only if we can read them!!
    c. **Group Activities are part of the graded portion of the lab manual!**
    d. Self check quizzes and practice material after the lab are optional.
    e. Students will supply completed grading rubric for grading

Proficiency Demonstrations:

- Every student in CS163 must show proficiency in programming in C++ and data structures, using linux with either vi, vim, or emacs,
- Demonstrating syntax proficiency is Pass/No Pass.
- Students will be asked to demonstrate linux editor commands for navigation, search and replace. This is Pass/No Pass.
- Demonstrations will occur twice a term.
- Proficiency demos are scored as:
    - E (exceeds, passing),
    - P (proficient, passing),
    - PW (pass with warning, passing)
    - IP (in-progress, non-passing),
    - U (unsatisfactory, non-passing)
    -
- A passing score is required to pass CS163.
- 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.
- If a student receives a non-passing score on the proficiency demos, they may meet with the instructor to discuss their performance.
- At the midterm time, students receiving an IP (in-progress) score may **retake the demo once** within a one week period. It is the student's responsibility to schedule the retake.
- There are no re-tests available for the final proficiency demo.

Computing Environment:

The **CS linux** systems using the g++ compiler **must** be used for all assignments and lab. Students must use the GNU GCC C++ compiler (g++) in the default –ansi mode. This mode meets the **C++ standard 98** guidelines (which can be obtained through these flags: –std=c++98, -std=gnu++98, or -ansi).

Students should compile with the –Wall flag to enable warnings about constructions that should be avoided. Students may compile with the –g flag to support the use of gdb. Use of any other flags is not allowed.

Students must use vi, vim, or emacs when programming in C++. No IDEs may be used for C++ coding; Dev Cpp, Visual C++, xcode are not acceptable

Each student needs a CS account (which can be received from FAB 88 or contacting the tutors (tutors@cs.pdx.edu)

- Lab work is done using **quizor2.cs.pdx.edu.**
    - Login with your CS account login name and password.
    - The lab system is only available for the term in which you take the class and all materials will be deleted at the end of the term.
- Individual graded programming assignments are to be done using **linux.cs.pdx.edu**

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.*

1. Every student in CS163 is enrolled in both a lecture and a lab session.
    a. Attendance **to the lab sessions is required to pass CS163**
    b. **Bring PSU Picture ID** (required to check-out a school computer).
    c. **If you miss more than one lab session, it needs to be made up within a 1-2 week period.** Schedule a makeup lab: https://cs163_makeup_lab.youcanbook.me/
2. **Bring both the Linux and Vim manual (for 2017-2018) and the CS163 Lab Manual for Spring 2018.** They may be purchased from the PSU Book Store.
    a. **Each student must have their own manuals.**
    b. **Lab manuals** will be collected and graded twice a term.
    c. **Lab manuals must be brought** to each lab session to attend.
    d. **Barcodes** will be attached to the first page of the manual which will be scanned on entry and exit of the lab to count as attendance. PSU ID's may also be used.
    e. **From the Linux and Vim manual, the Level #2 exercises must be completed by the end of the term.** Students who did not take PSU's CS162 course should also complete Level #1 exercises.
3. **Every Lab** will include exercises from both manuals.
    a. **Read** the background for each lab, **prior** to attending the corresponding lab!
    b. **Expect the lab assistants** to initial and date the completion of each lab and make corrections to prelabs within the lab manual.
    c. **Plan** to fill out all of the questions and progress through the lab manual as it is designed.
    d. **Make sure** to perform and practice all of the vim exercises as part of the lab.
4. Most lab sessions will be held in classrooms without computers.
    a. Therefore, it is important to either bring a laptop, netbook, or tablet to the lab session or **notify your instructor 24 hours in advance that a school computer is needed**.
    b. Make sure to fully charge your computer –electrical outlets may not be available.
    c. Power cords may not be stretched across walk-ways. Be aware of trip hazards.
5. Please be courteous to other people working and keep the volume low so that others can concentrate.
6. No food or drink while labs are in session
7. No use of the internet for web surfing, social media, or email during lab time. With the exception of D2L, karlaf's website, and the use of putty, ssh, or terminal to work remotely with the CS systems. *Any violation will result in an immediate No Pass for that particular lab session.*

Individual Programming Assignments:

1. **Individual Coding:**
   a. There are 5 programming assignments.
   b. They comprise 20% of your grade in the course.
   c. All programs must be created individually and written in C++ on CS linux using a linux editor.
   d. Every assignment MUST be backed-up prior to uploading to submitting.
   e. **NEVER** create an archive of your work without first backing it up!!!!!!

2. **Assignments consist of programs and their corresponding write-ups**
   a. **Due dates are specified in the course outline**
   b. **Assignments (Programs and Writeups)** may **not** be turned in later than the late due date***. There are no exceptions.***
   c. There are 4 written design write-ups that are turned in **prior** to the first four 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 D2L. **Design write-ups** and **test plans** are due prior to the program by **7pm.** The design write-ups must be a minimum of 600 words.
   d. **Late write-ups** will be accepted as specified by the course outline, by **7pm with a 5% deduction.**
   e. Be careful to not plagiarize. Doing so will result in a zero on an assignment and a failure in the class.
   f. **Every** write-up must be completed and submitted.

3. **Programs are due on uploaded to D2L by 7pm**. Plan ahead!
   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. **Late programs** will be accepted by **the specified late due date with a 5% deduction. Refer to the course outline for exact due dates.**
   c. **The last program (#5)** may NOT be turned in late.
   d. **Every** program must be completed and submitted.
   e. **Passing scores** are expected **on average** to pass CS163.
   f. **Programs incorrectly** uploaded (to the incorrect folder or unreadable) will receive a 5% penalty.

4. **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 "to much help" is not a valid reason to receive a passing score.
   b. Always make a **back-up** of your work. Deleting your work and asking for an extension is not acceptable. A back-up should <u>always</u> be made in a different folder or system.
   c. Never let anyone write the code for you! Never just copy code from the internet!

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

6. 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 place function implementations in a .h file and never #include a .cpp file!

7. **Submit assignments** using **tar** (or **zip**) electronically **to the D2L Dropbox** (make sure to select the "submit" button after uploading the files, otherwise your submission will be lost).
    a. **Before using tar or zip**, make sure to backup your files.
    b. **Check the archive**, to make sure that they are built correct and that you have not destroyed the contents of the file(s).
    c. **Check** D2L afterwards to make sure your files are correct.
    d. **Programs incorrectly** archived will receive a **5% penalty**.

8. **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. However, to pass the class the course you must have an average passing score on your programming assignments.

9. ***A score of 40% is required to pass and an overall average of all assignments must be 65% or greater to pass.***

Grading Policies:

1. Two quizzes combined are 5% of your grade
2. A Midterm Exam is 25% of your grade
3. The Final Exam is 40% of your grade

- **It is against department policy to give exams early (no exceptions!). Exams will all be closed book, closed notes.**

4. **All DRC exams must be arranged to take place at the same time as the in-**class exams, except when otherwise authorized by your teacher. Such authorization must be requested prior to 24 hours in advance of the in-class exam taking place.
   - **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, the midterm **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. Assignments will not be accepted after 1 week late. ***There are no exceptions.***
- Assignments will not be accepted after the last day of class.

- **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, not taken any exams, and have not participated on D2L. 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 <u>not</u> 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.

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 copies lab manual solutions from another student
6. Student accepts an assignment and/or program from another student
7. Student supplies an assignment and/or program to another student
8. Student posts the assignment and/or program on the web, social networking site, or D2L discussions
9. Student shares their password with another student at PSU giving that student access to their assignments and/or programs
10. Students work together on assignments and turn in the same and/or similar assignments.
11. Student turns in work that was obtained from other sources such as the web, friends, tutors or TA's.
12. Student leaves work available for others to copy from
13. Student attempts to purchase programs from others (in person or electronically).

Performing any of these actions will result in a ZERO grade on that assignment.

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 D2L discussions, the web, or social networking sites.
- Never give your assignments to any 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 when asking for help. Ask them to teach you what you need to know rather than do it for you!!!

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.