

**SYLLABUS**  
**CS 202 Programming Systems**  
**Spring 2018**

---

<b>Prerequisite:</b>	<b>CS163 Data structures</b> <i>Students are expected to be able to fluently write complete programs in C++ at the linux command line (without an IDE) using classes, pointers, dynamic memory, developing data structures such as Linear Linked Lists, Circular Linked Lists, Doubly Linked Lists, Arrays of Linked Lists, Linked Lists of Arrays, Trees, and Graphs.</i>
----------------------	---

---

<b>Instructor:</b>	<b>Karla Steinbrugge Fant,</b>
<b>E-mail:</b>	<b>karlaf@pdx.edu</b>
<b>Message Phone:</b>	<b>503-725-5394</b>
<b>Office:</b>	<b>Fab 120-19</b>
<b>Office hours:</b>	<b>Tuesday and Thursday 2:30-3:20 in FAB 120-19</b>

  

<b>Text:</b>	C++ Primer Plus, Stephen Prata Thinking in Java, Bruce Eckel
--------------	---

  

<b>Lab Manuals:</b>	<b>Required.</b> Purchase from the PSU Book Store The Linux and Vim Manual published for <b>2017-2018 academic year</b> and The CS202 Lab Manual published for <b>Spring 2018</b> <b>(earlier editions are not acceptable)</b>
---------------------	---

  

<b>Lecture Notes:</b>	Lecture notes and power point slides are on D2L
<b>PSU ID:</b>	Bring your PSU ID card to all lectures and labs; it will be used for attendance purposes.
<b>Handouts:</b>	All handouts, due dates, and assignments can be retrieved from D2L
<b>Disabilities:</b>	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 as close to the same time as the in-class quizzes and exams as possible, except when otherwise pre-authorized. Such pre-authorization should take place at least 24 hours prior to the in-class quiz or exam. <b>Quizzes and exams <u>may not</u> be taken prior to the in-class students.</b>

---

<b>System &amp; Compiler:</b>	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 <b>C++ standard 98</b> guidelines (-ansi). <b>Editors MUST be either: vi, vim, emacs</b> <i>No IDEs are allowed when working in C++;</i> <b>Expected: Use of gdb with each C++ assignment</b>
-------------------------------	---

---

**Prior Knowledge expected:**

CS202 is designed for students who have already have programmed in C++ and are proficient applying recursion to data structure algorithms. Students should already be able to program each of the following using C++ at the linux command line, without an IDE:

- (a) Classes, constructors, destructors
- (b) Pointers and dynamic memory (new, delete)
- (c) Pointer arithmetic
- (d) Dynamically allocated arrays
- (e) Linear, circular, and doubly linked lists
- (f) Arrays of linked lists
- (g) Binary search trees
- (h) Be prepared to apply algorithms to implement balanced trees and graphs

Students should be able to demonstrate proficiency at applying recursive solutions to insert, traverse or remove from pointer based linked data structures of: linear linked lists, circular linked lists, doubly linked lists, arrays of linked lists, and binary search trees.

**Course Description and Goals:**

Students will become familiar with the language and operating system environment used in most upper division courses in the Computer Science major curriculum. Use of the file system, operating system calls, and shell-level programming; low-level debugging of high-level programs. Programming exercises will include applications of data structures and memory management techniques.

The primary goal in CS202 is to prepare students for programming in the upper division 3xx and 4xx level classes. To achieve this goal, CS202 focuses on three areas: object oriented programming, advanced C++, and an overview of how Java relates to what we have learned in C++.

The majority of the term will be spent introducing students to object-oriented programming while learning advanced C++ syntax. Students will understand the difference between procedural abstraction and object oriented solutions. Students will spend the term designing and programming with inheritance hierarchies, with the goal of solving problems efficiently: producing high quality, robust, maintainable as well as efficient object oriented solutions. This will provide students with the chance to experience object oriented design and programming. Programming assignments will focus on advanced data structures while at the same time accomplishing these other goals. Students will learn about C++'s function overloading, operator overloading, copy constructors, and be introduced to inheritance.

Java skills developed include writing two programs using advanced data structures in Java with strict requirements to follow OOP guidelines – all data members private, no friendly access, and complete implementation of functions required to handle issues of deep versus shallow copies and compares. Students learn the relationship between the two languages and the similarity of Java's references to pointers.

## 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 each lab session.
  - 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 session. If lab work is completed prior to the end of the lab period, practice questions may be assigned by the lab assistants.
  - f. Students are encouraged to continue working on lab materials after lab sessions have concluded.
3. **If you miss more than one lab, it needs to be made up within a 1-2 week period.**
  - a. Attendance to a makeup lab must be scheduled through [https://cs202\\_makeup\\_lab.youcanbook.me/](https://cs202_makeup_lab.youcanbook.me/)
  - b. To makeup **more than two** labs, contact [karlaf@pdx.edu](mailto:karlaf@pdx.edu) to seek authorization.
  - c. To makeup a lab beyond the 1-2 week time period, contact [karlaf@pdx.edu](mailto: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 completion and readability. Answers only count if we can read them!!
6. **Assignments consist of programs and their corresponding write-ups**
  - a. **Due dates are specified in the course outline**
  - b. **OO write-ups** and **UML** diagrams are due prior to the program by **6pm**
  - c. **Late write-ups** will be accepted by late due date **by 6pm**
  - d. Be careful to not plagiarize. Doing so will result in a zero on an assignment and a failure in the class.
  - e. **Every** write-up must be completed and submitted.
7. **Programs due dates**
  - a. **Late programs** will be accepted by **the specified late due date with a 5% deduction. Refer to the course outline for exact due dates.**
  - b. **The last program (#5)** may NOT be turned in late.
  - c. **Every** program must be completed and submitted.
  - d. **Passing scores** are expected **on all** to programs to pass CS202.

### Overview of Grading Policies:

Demonstrate Proficiency in C++ <ul style="list-style-type: none"><li>- Midterm Demo</li><li>- Final Demo</li></ul>	Pass/No Pass	At PSU or by Proctor (must pass both demos)
Lab Participation <ul style="list-style-type: none"><li>- Prelabs</li><li>- Lab Code Submitted</li></ul>	Pass/No Pass	Pre-labs and Lab code (Attendance to all but 1 lab required)
Graded Lab Manuals <ul style="list-style-type: none"><li>- Student will submit a rubric with the manual</li></ul>	5%	Lab manuals are graded twice (Manuals are graded for completeness, relevance, and readability)
Individual Assignments <ul style="list-style-type: none"><li>- Term Paper</li><li>- 3 Written Designs/UML</li><li>- 5 Programming Projects</li></ul>	5% 5% 20%	Submit to D2L Dropbox (All designs, UML diagrams, programs, and papers must be submitted and each receive a passing score of 65% or greater)
Quizzes and Midterm Exam <ul style="list-style-type: none"><li>- 2 Quizzes</li><li>- Midterm Exam</li></ul>	5% 25%	At PSU or by Proctor (The midterm score must be 65% or greater to pass CS202)
Comprehensive Final Exam	35%	At PSU or by Proctor <b>*** Must receive a Passing score of 65% to pass the class ***</b>

### Proficiency Demonstrations:

- 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 proficiency demo taken during the middle of the term may be retaken within one week to receive a passing score **with approval by the instructor**.
  - Midterm proficiency demos may only be retaken once. Retakes need to be taken with a 1-2 week period.
  - There are no re-tests available for the final proficiency demo.
- A passing score is required to pass CS202.
  - **Final proficiency demonstrations are only scored for students receiving a passing grade in CS202.**
  - If a student receives a non-passing score on the proficiency demos, they may meet with the instructor to discuss their performance.
  - Every student in CS202 must show proficiency in programming in C++ and data structures, using linux with either vi, vim, or emacs, and gdb. Students in CS202 are **expected** to be fluent with vim (or emacs). This is Pass/No Pass.

### **Lab Sessions:**

The labs 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 perform the lab work each week.*

- Every student in CS202 is enrolled in both a lecture and a lab session.
- Attendance **to the labs is required to pass CS202**
- **Bring PSU Picture ID** (required to check-out a school computer).
- **Bring both the Linux and Vim manual (for 2017-2018) and the CS202 Lab Manual for Spring 2018.** They may be purchased from the PSU Book Store.
- **Each student must have their own manuals.**
- **Lab manuals** will be collected and graded twice a term.
- **Lab manuals must be brought** to each lab session to attend.
- **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.
- **From the Linux and Vim manual, the Level #3 exercises must be completed by the end of the term.** Students who did not take PSU's CS162/163 courses should also complete Level #1 and Level #2 exercises.
- **Read** the background for each lab, **prior** to attending the corresponding lab!
- **Expect the lab assistants** to initial and date the completion of each lab and make corrections to prelabs within the lab manual.
- **Plan** to fill out all of the questions and progress through the lab manual as it is designed.
- **Every Lab** will include exercises from both manuals.
- **Make sure** to perform and practice all of the vim exercises as part of the lab.
- **If you miss more than one lab, it needs to be made up within a 1-2 week period.** Schedule a makeup lab: [https://cs202\\_makeup\\_lab.youcanbook.me/](https://cs202_makeup_lab.youcanbook.me/)
- Most lab sessions will be held in classrooms without computers. 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.
- Make sure to fully charge your computer –electrical outlets may not be available.
- Power cords may not be stretched across walk-ways. Be aware of trip hazards.
- No food or drink while labs are in session
- 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:**

- **Every assignment (programs and write-ups) must be submitted and receive a passing grade** to pass CS202.
- 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.
- Every assignment must be **backed-up** before creating an archive. A back-up should always be made in a different folder or system. If your code become corrupt due to the archive process, there will be a 5% penalty.

- **Due dates** and late dates are specified in the **Course Outline**.

- **Assignments (Programs and Writeups)** may **not** be turned in later than the late due date. *There are no exceptions.*
- **Late** programs and write-ups will have a 5% automatic deduction.
- **Due “times”** are always by **6pm on the due date or late date**. Plan ahead!

- **Design Write-ups and UML diagrams:**

1. There are 3 written design OO write-ups that are turned in **prior** to the related program, with accompanying UML diagram. The diagrams may be hand-drawn (and scanned) or electronically generated and submitted as a pdf.
2. **All written design write-ups must be completed** to pass CS202.
3. **They must consist of a minimum of 600 words**
  - The write-up should cover the major design considerations: what classes you are using, how they are related to other classes (using, containing, hierarchical); it should discuss how the methods provided minimize the use of getters. It should be clear in the design what functions will be needed in each class and how they will be used by the other classes in your design.
  - All designs 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.
  - *Refer to the background information sheet on D2L for details.*

- **Individual Coding:**

1. There are 5 programming assignments.
2. They comprise 20% of your grade in the course.
3. All programs must be created individually and written in C++ on CS linux using a linux editor.
4. The programming assignments provide experience building correct implementations of object oriented situations.
5. **All 5 programs must be submitted and received a passing score to pass CS202**

6. **Each programming assignment must include the following write-ups (turned in at the same time as the program):**
  - A minimum 400 word written document analyzing your design and discussing how effective the classes that you created were. Discuss the validity of your approach and analyze it in terms of object oriented programming. **Think in terms of analyzing your solution!** This means discussing the efficiency of the approach as well as the efficiency of the resulting code.
  - A minimum 200 word written discussion of how debuggers (gdb, xxgdb, ddd, etc.) assisted in the development. This write up must describe experiences with the debugger, how it assisted code development, or how it could be used to enhance the programming experience.
7. **Focus on program style and comments:**
  - Each file must have your name and header comments describing the purpose of the code within the file.
  - The file header comments should be a **paragraph**, at a minimum
  - Each function must have header comments describing the purpose of the function and arguments. No exceptions!
  - Make sure that each function that has a non-void return type returns a value through each possible path
  - Always use the returned value when calling a function
  - Avoid single character variable names, except for loop control variables and array indices
  - Avoid while(1) or the use of break from within a loop
  - Very few functions in CS202 should have void return types.
8. 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!
  - Limit your submissions to **FIVE** source code files in total.
9. **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).
  - **Before using tar or zip**, make sure to backup your files.
  - **Check the archive**, to make sure that they are built correct and that you have not destroyed the contents of the file(s).
  - **Check D2L afterwards** to make sure your files are correct.
10. **Due “times” are always by 6pm on the due date or late date.**

### 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](mailto:tutors@cs.pdx.edu)))
- Lab work is done using **quizzor2.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**

### Term Paper:

- Each student will be required to submit a typed term paper. The paper must **explore how well the programs written this term have met our objectives of being object oriented! Discuss how the designs met the criteria set out for OOP, and how they could have been improved.**
- The paper must be a minimum length of 4 pages and a maximum of 7 pages (double spaced, 12 point font) of text. This does not include the cover sheet, table of contents or reference pages.
- Attach any tables or sample code as exhibits; these may not be part of the 4-7 page count.
- The term paper should be done using **Word** or plain text. It may be submitted as a pdf if other word processors are used.

### Grading Policies:

- Two quizzes combined are 5% of your grade
- A Midterm Exam is 25% of your grade
- The Final Exam is 35% of your grade

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

- 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**, term paper, 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.***



- **All DRC exams must be arranged to take place at the same time as the in-class exams** whenever possible, 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.
  - **Exams at the testing center can never be taken before the in-class students take their exam.**
- **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 not expect an incomplete in this class.***

### **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.
- Never have another person use your keyboard to type in code for you. Have them teach you what is needed rather than take over!!!

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.

### **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 and reported to the Dean of Students.

#### **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.**

***Remember to Program Every day!!***