Winter 2020 Syllabus

Instructor: Paul Roper	**Section 001**	**Section 002**
Office: 3370 TMCB, 422-8149	**10:00-10:50 am**	**11:00-11:50 am**
Hours: 9:00-9:50 am MWF	**MWF**	**MWF**
Email: proper@cs.byu.edu	**1170 TMCB**	**1170 TMCB**
Instructor: Parris Egbert	**Section 003**	**Section 004**
Office: 3318 TMCB, 422-4029	**8:00-9:15 am TTh**	**9:30-10:45 am TTh**
Hours: By Appointment	**1170 TMCB**	**1170 TMCB**
Email: egbert@cs.byu.edu		

Learning Outcomes

CS 235 provides a hands-on introduction to data structures and algorithms. Through programming projects and exams, you learn how to select and implement suitable data structures for various problems, how and when to use recursion, and how to analyze the cost of algorithms, especially sorting and searching algorithms.

- **Data Computing Fundamentals** - Implement the fundamental data types of computing (lists, stacks, queues, heaps, priority queues, sets, maps, and binary search trees).
- **Select Data Structures** - Properly use and select data structures from language-provided data-structure libraries.
- **Analysis** - Apply basic algorithm analysis.
- **Recursion** - Understand how recursion works and write programs using recursion to solve problems.
- **Choice of Algorithms** - Make informed decisions about which sorting and searching algorithms to use in specific circumstances.
- **Program Writing** - Write larger, multi-file programs using an IDE.

Pre-requisites

- Students must have completed CS 142 (or an equivalent) with a passing grade before taking CS 235.
- Students cannot retake CS 235 for a 2nd time without fulfilling the CS department retake policy.
- **If you have not satisfied any of the above pre-requisites, you may be dropped from the course.**

Course Materials

All course content including syllabus, schedule, PowerPoint slides, lab descriptions/submission/review, attendance quizzes, and help queue are available https://students.cs.byu.edu/~cs235ta . Learning Suite **will only be used for lab, review, quiz, and exam scores** under the BYU Grades tab.

C++ is a standard, not a program. Within the functionality and syntax defined by C++ standards committees, your source code should compile and execute independent of what compiler or IDE you choose (gcc, g++, MSVC, LLVM, etc.) As such, we will be using Microsoft's Visual Studio Community C++ IDE in class and g++ for lab submissions.

You will greatly benefit from using IDE based operations of setting **breakpoints**, **single stepping**, and **memory inspection** during the debug process. All CS department machines have Visual Studio installed. You can find Visual Studio's C++ language references here: https://msdn.microsoft.com/en-us/library/3bstk3k5.aspx. (A free install is available for all Windows machines at http://www.visualstudio.com/en-us/downloads/download-visual-studio-vs#DownloadFamilies_2.)

We will use Elliot B. Koffnam and Paul A. T. Wolfgang's Objects, Abstraction, Data Structures and Design Using C++ for out lectures. Our course material follows the order of the book chapters.

Finally, make www.stackoverflow.com, http://www.cplusplus.com/reference and www.google.com your closest friends. If you have any questions about C++, consult them first before seeking TA help.

PowerPoint lectures and class discussions will be each class period as outlined on the course schedule. There will be times when information given in class is not found in the text book or on the class website. It is very important that students attend class and take careful notes. You will be held responsible for class material as well as the course text.

To encourage class attendance and to help prepare you for the exams, an on-line quiz will be available for each lecture. Codes for the attendance quizzes will be announced during class and expire at the end of the week.

PLEASE AVOID USING LAPTOPS/CELL PHONES DURING CLASS TIME. While there are arguable benefits to having computers available in class, our class rooms are full and electronic media usage during lectures is distracting to others and does not help you focus on the discussion. Many studies clearly show those who use computers during class time, even just for note taking, receive lower grades. So, break out the pencil and paper, invest in a paper notebook, and come prepared to participate in class.

Computer Programming Assignments

Programming assignments consist of 11 labs designed to emphasize topics and objectives discussed in class.

- **Lab 1: Grades**. Output the average score for all students on each exam followed by a list of student's score and grades.
- **Lab 2: SNAP**. Design classes for a school database and output course and student schedules.
- **Lab 3: Linked List.** Develop a Linked List container that implements standard list operations.
- **Lab 4: Iterator (linked list)**. Implement a nested iterator class for Lab 3's linked list container.
- **Lab 5: Expressions (infix, postfix, balance).** Performs operations on infix, postfix, and prefix expressions.
- **Lab 6: Railyard (stacks, queues, deques)**. Manage the Greater Provo Railroad Station storage facility as directed.
- **Lab 7: Maze (recursion).** Two SCUBA divers need to find a path through the storage facility maze.
- **Lab 8: BST.** Add and remove nodes from a binary search tree.
- **Lab 9: Pokémon (maps and sets).** Add and remove nodes from map and set data structures.
- **Lab 10: Quicksort.** Implement a quicksort algorithm and sort a data set.
- **Lab 11: AVL (balanced tree).** Use AVL algorithm to self-balance a binary search tree.

Please pay careful attention to the following programming assignments guidelines:

- Zip together all your lab source files (.cpp, .hpp, .h) – do not include any IDE folders or files.
- Submit your zipped lab source folder using the “Labs” tab on the class website.
- Your program must use command line arguments to select both input and output files.
- The auto-grader will unzip and compile your submitted source using the Linux g++ compiler.
- The resulting executable will be graded using lab test input files and the resulting output files.
- The results of each test will be reported immediately on the submission web page.
- Points will be deducted for any detected memory leaks, warnings, and/or out-of-bounds array references.
- You may re-submit your zipped source as many times as you like throughout the semester. (Late penalties may apply.) Your highest score will be used for your final lab grade.
- Anonymous peer review assignments will be made immediately after the lab submission deadline.
- Assigned peer reviews must be completed before the next lab deadline.
- Lab and peer review scores will be recorded weekly in BYU grades.
- Only on-time zipped solutions of more than 10 points will participate in the peer review process.
- **Lab assignments submitted after the due date will receive a 50% penalty on portions of the lab not submitted ontime.**
- **It is the student’s responsibility to ensure all relevant files are included in the zipped submission** - submissions that don’t compile or are from the wrong lab will result in a lower grade.
- Lab grades will be computed as follows:
  - 30-50 points - Lab requirements assessed by auto-grader.
  - 10-20 points – Coding style and additional requirements assessed by peer reviewers.
  - 10 points – Completion of two assigned peer reviews (5 points each.)

Note:

- To receive an A grade for the course, you must complete (fully functional) all 11 required labs and score 70% or better on the final exam.
- If you want your work to remain anonymous during peer reviews (FERPA), use a unique coding pseudo-name (e.g. Cody Coder III, Franky Forloop, Doesy Dowhile) on source headers comments. Using identifying comments serves as your permission for your peers to know your work’s identity.
- Peer review scores that significantly differ will automatically be arbitrated by a TA. Submitting inaccurate peer reviews will result in loss of your review points.

Exams

A pre-exam is required of all students and is administered at the beginning of the semester in the Testing Center. Your score on the exam will not affect your final grade.

One midterm will be administered during the semester in the Testing Center as scheduled. You should plan at least 1-2 hours for the exam. Any exam taken after the scheduled exam days will be worth (maximum) half credit unless prior arrangements are made with the instructor.

The final exam is comprehensive and will also be administered in the Testing Center, during finals week (not during LS assigned time in class.) There is no makeup for the final exam.

Quizzes (5% Extra credit)

Attendance quizzes are available on the class website during the week of the associate lecture. The quiz code will be given out during class time and expires at the end of the week. Quizzes assess material covered in class and the book and help prepare students for the exams.

There is **no makeup** for missed quizzes, but the quiz material (**without the answers**) will be available in the class PowerPoint slides after the quiz code expires. Sharing of quiz codes is considered cheating.

Grades

Your grade will be based on programming assignments (60%), exams (40%), and attendance quizzes (5% extra credit) as follows:

- Programming assignments (60%)
    - Lab 1: Grades – 4%
    - Lab 2: SNAP – 5%
    - Lab 4: Linked List – 5%
    - Lab 3: Iterator – 5%
    - Lab 5: Expressions – 5%
    - Lab 6: Railroad – 6%
    - Lab 7: Maze – 6%
    - Lab 8: BST – 6%
    - Lab 9: Pokémon – 6%
    - Lab 10: Quicksort – 6%
    - Lab 11: AVL – 6%
- Pre-exam – 5%
- Midterm – 15%
- Final – 20%
- Attendance Quizzes – (5% Extra credit)

Grading Scale:

100-93	A	82-80	B-	69-67	D+
92-90	A-	79-77	C+	66-63	D
89-87	B+	76-73	C	62-60	D-
86-83	B	72-70	C-	59-0	E

Miscellaneous

Do's:

- Do have a valid email address and check your email often.
- Do complete your work on time. Partial credit will be awarded. No late work accepted.
- Do discuss labs and course material with your classmates and study groups, but do your own work.
- Do use TA time for specific questions.
- Do break out the pencil and paper and come prepared to participate in class.

Don'ts:

- DO NOT post your work on a public repository (e.g. github, Learning Suite, Bitbucket, etc.)! We troll often – posting lab solutions, exam preps, and quiz answers will result in a failing grade for the course. Private repositories are encouraged.
- DO NOT submit someone else's work. Submitting others' work is considered cheating for which you will receive an E grade for the class and be reported to the Honor Code Office. Your source submissions will be compared with past and current submissions. No grade is worth the consequences of dishonesty!
- DO NOT use your laptops or other electronic media during class – break out the pencil and paper.

Grading Verification and Appeal Policy

It is the student's responsibility to verify that lab, homework, and exam scores have been correctly recorded in Gradebook **before** the last day of class. Please disregard Gradebook's grading system (ie, final grade, maximum points, etc.) until further notice.

Academic Honesty

The Honor Code includes a statement of standards regarding academic honesty. Academic honesty includes completing your own homework, programming assignments, midterms, and final. Students should work together to help each other understand material, but should always turn in their own work. Examples of academic dishonesty include sharing code for assignments with other students, turning in someone else's code as your own, sharing exam material with other students, and cheating on an exam. Any violation of academic honesty standards will result in failing the class. All violations of academic honesty are documented and reported to the Honor Code office.

Posting, downloading, or viewing any classwork on a public repository such as GitHub is considered cheating. This policy applies even after this class ends – grades can be changed at any time!

Preventing Sexual Harassment and Discrimination

Title IX of the Education Amendments of 1972 prohibits sex discrimination against any participant in an educational program or activity that receives federal funds. The act is intended to eliminate sex discrimination in education. Title IX covers discrimination in programs, admissions, activities, and student-to-student sexual harassment. The BYU policy against sexual harassment extends not only to employees of the university but to students as well. If you encounter unlawful sexual harassment or gender based discrimination, please talk to your professor; contact the Equal Employment Office at 378-5895 or 367-5689 (24-hours); or contact the Honor Code Office at 378-2847.

Students with Disabilities

Brigham Young University is committed to providing a working and learning atmosphere which reasonably accommodates qualified persons with disabilities. If you have any disability which may impair your ability to complete this course successfully, please contact the Services for Students with Disabilities Office (378-2767). Reasonable academic accommodations are reviewed for all students who have qualified documented disabilities. Services are coordinated with the student and instructor by the SSD Office. If you need assistance or if you feel you have been unlawfully discriminated against on the basis of disability, you may seek resolution through established grievance policy and procedures. You should contact the Equal Employment Office at 378-5895, D-282 ASB.

Disclaimer

This syllabus is subject to change during the semester in favor of the student as deemed necessary by the instructor. It is the student's responsibility to stay current with class policies and check the class website, Learning Suite, and their email often during the week.