CS 215 – Fundamentals of Programming II

Fall 2018 – Syllabus

Instructor

Dr. Don Roberts
Email: roberts@refactory.com or roberts@evansville.edu
Web: `http://csserver.evansville.edu/~droberts`

Class Hours

MTTh, 2:10PM–3:00PM, Room 130

Office Hours

M, 3:00 PM–4:00 PM T, 9:45 AM–10:45 AM or by appointment

Catalog Data

Project and problem-solving course emphasizing the use of classes for encapsulation of abstract data types and abstract data structures. Topics include classes, templates, dynamic allocation, searching and sorting, recursion, and exception handling.

Objectives

The main objective of this course is to continue the study of problem solving techniques used in programming software solutions with emphasis on abstract data types, and to become proficient in the use of the UNIX operating system and development tools. Specific outcomes for this course include:

- Students will be able to write programs using selection and repetition constructs, functions, and arrays.
- Students will be able to use dynamic allocation and recursion to solve problems.
- Students will be able to use design and implement abstract data types (ADTs) using simpler data structures such as multi-dimensional arrays and vectors, and use these ADTs to solve problems.
- Students will be able to design and implement abstract data types (ADTs) using linked data structures such as linked lists and binary trees, and use these ADTs to solve problems.
- Students will be able to implement several sorting algorithms.
- Students will be introduced to rudimentary algorithm analysis.
- Students will be able to do basic generic programming and use library structures such as stacks and queues to solve problems.
- Students will complete at least 4 assignments requiring design of an application using one or more data structures, testing, and debugging.
- Students will be proficient using the UNIX operating system.

- Students will be proficient using a UNIX editor, such as emacs or vim.
- Students will be proficient using relevant GNU tools for software development, including g++ (compiler) and make (program build utility).
- Students will be introduced to contemporary professional issues.

Prerequisites

Grade of C- or better in CS 210

Required Textbook

Wittenberg, *Data Structures and Algorithms in C++,* Mercury Learning and Information, LLC, 2018, ISBN:978-1-68392-0840-7.

Daily and Weekly Requirements

Assigned daily reading and weekly homework assignments. Homework assignments may include short programming problems.

Programming Projects

There will be 7–9 programming projects of 1–2 weeks in duration each. See handout *A C++ Programming Guideline for CS 215* for the appropriate code format to use for this course.

For each project, the grade will be 75% for correctness and 25% for style/design.

Exams and Evaluations

There will be one in-class written exam and a comprehensive final written exam. In addition, there will be two, 2-hour programming practical exams. The purpose of the practical exam is to demonstrate mastery in using the C++ programming language and the UNIX environment. Therefore, it is necessary to score a minimum of 60% on the second practical exam to pass the course. Students who fail to do so and are otherwise passing, will be given a second opportunity to pass the practical exam at the end of the term with a 10% penalty. Final grades will be based on the following distribution:

20 %	Comprehensive Final Exam
15 %	in-class exam
20 %	Lab programming practical exams
5 %	Homework
40 %	Programming Projects

Attendance, Missed Exams, Late Homework and Machine Problems

Homework problems are due at midnight on the date specified. Machine Problems are due at midnight on the date specified. Any assignments handed in after that time are considered late. The following late penalties will be applied:

One day late	10%
Two days late	10% + 20% = 30%
Three days late	10% + 20% + 30% = 60%
Four days late	Do the math...

Valid excuses for missing exams and handing assignments in late include illness, family emergencies, religious observances, official UE events, etc. They do not include work conflicts, studying for

other classes, staying home an extra day over the weekend, or {dog, virus, space aliens}{ate, wiped out, abducted}{my homework, my computer, me} (in any combination).

The instructor will rely on your integrity on getting work excused. If you have a valid excuse, put it in writing, sign your name to it, and give it to the instructor. For religious observances and UE events, you must inform the instructor that you will be absent **before** the absence occurs, otherwise it will be considered an unexcused absence.

Excused work must be made up within two class meetings. Late work will not be allowed. Exceptions will be made for serious or prolonged illness, or other serious problems. Please note that it is your responsibility to take care of missed or late work.

Honor Code

All students are expected to adhere to the University's Honor Code regarding giving and receiving authorized aid.

- **Homework** exercises are for you to gain experience and practice. You may collaborate with your classmates, but each student should submit a solution in his/her own words that reflect his/her understanding of the solution. You will eventually be required to demonstrate your knowledge of the material on the exams, so it is better that you attempt the problems on your own.
- **Programming projects are to be your own work.** Solutions shall not be copied from Internet or other sources. Discovery will result in a 0 for that assignment for all parties involved for the first offense. A second offense will result in failure of the course.
- **You may not observe another person's code or solution**. You are expected to design and code your solution yourself. UNIX systems tend to be open by default so it is absolutely forbidden to "rummage" around the file system looking at anyone else's work even if they have not set the file permissions correctly.
- *Any* attempt to compromise the automatic grading system will result in immediate failure of the course.

Disability Policy

It is the policy and practice of the University of Evansville to make reasonable accommodations for students with properly documented disabilities. Students should contact the Office of Counseling and Health Education at 488-2663 to seek services or accommodations for disabilities. Written notification to faculty from the Office of Counseling and Health Education is required for academic accommodations.

Schedule

Wk	Date	Monday	Tuesday	Thursday	Friday
1	*[2018-08-27 Mon]*	Overview, basic UNIX, clang, console I/O, namespaces	Filestreams, `argv`, `argc` Version control, `git`	Reference Parameters Output formatting	
2	*[2018-09-03 Mon]*	Strings	Ch. 5.3–5.5 Recursion	Ch. 5.3–5.5 Recursion	Ch. 1.2 Classes
3	*[2018-09-10 Mon]*	Separate Compilation `make`	Ch. 1.2 Overloaded operators	Assertions Exceptions	
4	*[2018-09-17 Mon]*	Ch. 1.5–1.6.1, 3.1 Templates, STL Vector STL list	Ch. 1.1, 1.6.2: Iterators `auto` range-based `for` loops	**Practical Exam I** **Time: TBA**	
5	*[2018-09-24 Mon]*	Chap 1.6.3 STL algorithms	Ch. 1.3 pointers dynamic arrays	NO CLASS	
6	*[2018-10-01 Mon]*	Ch. 1.7.1–1.7.2 Vector Implementation	Review	**Midterm Exam**	
7	*[2018-10-08 Mon]*	Testing, debugging, `gdb`	Ch. 2, 5.6 Algorithm Analysis	NO CLASS	
8	*[2018-10-15 Mon]*	Ch. 3.1–3.2 Linked lists	(ON WEDNESDAY) Ch. 3.1–3.2 Linked Lists	NO CLASS	
9	*[2018-10-22 Mon]*	Ch. 3.2–3.3 Linked Lists	Ch. 4.1 STL stack and queue	Ch. 4.2 Stack Applications	
10	*[2018-10-29 Mon]*	Ch. 6.1 Binary Tree	Ch. 6.2 Binary Tree Applications	Ch. 6.3 Binary Search trees (BST)	Ch. 6.7.1–6.7.2 BST implementation
11	*[2018-11-05 Mon]*	Ch.6.7.2–6.7.3 BST implementation	Ch. 6.4–6.5 STL set and map	NO CLASS	
12	*[2018-11-12 Mon]*	**Practical Exam II** **Time: TBA**	Ch. 7.3 Heaps	Ch. 7.4–7.5 Priority queue	
13	*[2018-11-19 Mon]*	Chap. 8.5 Heapsort	Ch. 8.1–8.3 Bubble, selection, insert sort	Ch. 8.7 Quicksort	
14	*[2018-11-26 Mon]*	Ch. 8.6 Merge sort	Ch. 8.8: STL sort Comparing sorts	Final Exam Review	