Course Policies: CS215 Spring 2012
Introduction to Program Design, Abstraction and Problem Solving

1. Introduction

This course introduces you to object-oriented design and problem solving. Course subjects include data structures, dynamic data and pointers, and recursion. There is an introduction to sorting, searching, and the complexity of algorithms. Prerequisite: CS-115.

This course will meet on Tues/Thurs for lecture at 9:30am in Chemistry Physics (CP) room 155 (sections 001, 002, 003, 004, 005, 006, 007, 008). The evening section of the course (section 401) will meet on Tues/Thurs at 6:00pm in FPAT 255. Times for lab sessions for the day lecture are on Wednesdays and Fridays, and depend on the section in which you are registered. According to the University of Kentucky's course schedule guide, lab session meeting times are as follows:

Section	Teaching Assistant	Lecture Time / Place	Lab Time / Place
001	Chris Parks	Tues/Thurs 9:30am – 10:45 (CP 155)	Wed 8:15am – 9:45 (103 RGAN)
002	Xiwei (Jeffrey) Wang		Wed10:00am–12:00 (103 RGAN)
003	Xiwei (Jeffrey) Wang		Wed 12:00 – 1:30 (103 RGAN)
004	Zhenghao (Frank) He		Wed 2:00pm – 3:30 (103 RGAN)
005	Chris Parks		Fri 8:15am – 9:45 (103 RGAN)
006	Pengpeng (Mark) Lin		Fri 10:00am–11:30 (103 RGAN)
007	Pengpeng (Mark) Lin		Fri 12:00 – 1:30 (103 RGAN)
008	Zhenghao (Frank) He		Fri 2:00pm – 3:30 (103 RGAN)
401	Sami Taha	Tues 6pm–8pm (255 FPAT)	Thurs 6pm – 8pm (103 RGAN)

1.1 Professor and Teaching Assistants

The professor for this course is Dr. Brent Seales (http://dmn.netlab.uky.edu/~seales). The course homepage is http://dmn.netlab.uky.edu/~seales/cs215.html. Office hours: 2-4 Tues/Thurs. Office location: 102G Marksbury Building (http://ukcc.uky.edu/cgi-bin/dynamo?maps.391+campus+0633). The teaching assistants assigned to this course are Chris Parks (chris.parks@uky.edu), Sami TahaAbuSnaineh (sstaha2@uky.edu), Xiwei (Jeffrey) Wang (J.Wang@uky.edu), Pengpeng (Mark) Lin (pli222@uky.edu), and Zhenghao (Frank) He (zhenghao.he@uky.edu).

1.2 Course Materials

Course materials (assignments, problem sets, additional readings) will be available on line through the course web page. You are responsible for obtaining a copy of the primary textbook. The course is designed with the textbook as an integral part, and it is therefore expected that you will purchase your own copy for use in the class. The course web page and direct email will be the primary method of distributing information for the course. The **required text** for the course is *C++ For Everyone* (Second Edition), by Cay Horstmann (Wiley, ISBN 97-0-470-92713-7).

2. Grading and Assignment Policies

Your overall grade in CS215 will be determined according to relative weights below, with a letter grade based on overall performance assigned as indicated:

Programming Assignments:	30%	90-100	**A**
Exercises and Problem Sets:	30%	80-89	**B**
Quizzes:	10%	70-79	**C**
Midterm Exam:	15%	60-69	**D**
Final Exam:	15%	0-59	**E**

You will be required to complete three kinds of individual assignments using the C++ programming language: Programming Assignments, Individual Exercises, and Laboratory Exercises. The assignment sheets for this work will be available for download on the class web page.

Programming Assignments: There will be five programming assignments. These assignments are due as noted on the syllabus. You will be given an individual specification sheet for each assignment.

Individual Exercises: Each week you will complete assigned exercises, which will lead you through concepts presented in lecture material and are integrated with techniques to be used in programming assignments. Weekly assignment sheets will be available on-line the Friday before the week the material is due.
Laboratory Exercises: Laboratory exercises are to be completed and submitted in lab sessions, supervised and directed by your teaching assistant.

Individual exercises and lab exercises will be included all together on the *weekly assignment sheet*, available to you for download the Thursday before the week you will be doing that work. This will give you the weekend to start by reading and understanding what will be expected for the coming week.

The Midterm Exam will be given in class as scheduled on the syllabus. The Final Exam (cumulative) will be given in the room where you normally meet at the time scheduled below in accordance with the Final Exam Schedule for Spring 2012 issued by the University of Kentucky Registrar:

CS215 Section	**Final Exam Time**
001, 002, 003, 004, 005, 006, 007, 008	Friday 4 May 10:30am
401 (night section)	Tues 1 May 6:00pm

2.1 Deadlines and Submission Policies

All assignments and exercises must be submitted electronically. You will receive submission instructions in class and will find a link to the electronic submission page on the course web page. Excused absences (medical excuse requires Physician's written justification) will be handled on a case-by-case basis. The policy for submission of your work is as follows:

Programming Assignments: Due at *midnight* on the due date listed in the specification sheet and shown on the syllabus. Late work will not be accepted. Again, no late work will be accepted.
Individual Exercises: Due at *midnight* on due date. Individual exercises for the week will **always** be due on Thursday of that week. Late work will not be accepted. Again, ***no late work will be accepted***.
Laboratory Exercises: Due at midnight at the end of your lab day. Attendance at lab session is ***required*** in order to submit work. If you do not attend your scheduled lab session your laboratory exercises will receive no credit and will not be reviewed. Late lab submissions will not be accepted. Again, no late laboratory sessions will be accepted.

All submissions must conform to the platform requirements directed in class. Specifically, program submissions must be source code submissions that will compile and run using the integrated development environment IDE prescribed (Netbeans / Qt). Laboratory exercises must be submitted according to guidelines provided in your laboratory session.

2.2 Grading Policies

Feedback and evaluative benchmarks are an important part of understanding how to perform at the highest level of excellence in this class. Your work will be graded as follows:

Programming Assignments: Each submitted programming assignment will be compiled from source code and executed with public and private data to evaluate correct execution. Furthermore, the structure of your source code and its design, adherence to coding principles, and correctness for cases that may or may not appear in test data, will be judged during the grading process. Grades for programs will be assigned according to a five-point scale: 5 – excellent work; 4 – very good work; 3 – good, but some deficiencies; 2 – fair, has major deficiencies; 1 – poor; 0 – nothing submitted.

Program re-submission policy: Programs receiving less than a score of 5 (excellent) may be corrected and re-submitted. The final score for the assignment will be the average score of the original evaluation and the score of the corrected version. Deadlines for re-submission will be midnight Sunday following the day your assignment is returned to you. Since graded assignments will be returned in your lab session, this means that the re-submission deadline will always be the Sunday following the lab day (Wednesday) when your work is returned. Re-submission is not allowed if nothing was originally submitted, i.e., re-submission enables a **correction** of a submitted program, not an original submission.

Individual and lab exercises: Exercises will be graded on a "check/minus" basis. A correctly completed exercise will earn a "check" (full credit). A deficient assignment (feedback will be provided) will earn reduced credit. Exercises may not be re-submitted.
Quizzes: Weekly quizzes will be taken during the laboratory period and will be graded on a 10-point scale. No make-up quizzes are allowed.
Exams: The two exams (midterm and final) will be given during established times (listed on syllabus). No make-up exams will be allowed for unexcused absences.

2.3 Academic Dishonesty

Individual work (programming, appropriate lab exercises, exams) must be your own. You may discuss ideas with others, but no sharing of computer code or other work will be allowed. The University of Kentucky's guidelines regarding academic dishonesty will be strictly enforced.

3. Computer Facilities

We will be using the personal computers in RGAN 103 and University AD accounts. Treat your account as you would your personal and confidential records. You must have your University account set up in order to participate in the labs. During the first lab session, the TA will ensure that you can log into the machines in the lab. You are encouraged to use your own computer system for developing and testing your work (outside of lab). There are on-line instructions for installing and running Netbeans and Qt, which are platform-independent tools and will work on Windows, MacOS, and linux. You are responsible for configuring your own development environment if you choose to use your own computer to prepare and submit work.

4. In-Class Engagement

It is your responsibility to attend class consistently (lectures and labs) and remain engaged during those periods of time. Engagement means that you are paying attention to the class/lab activity. It means that you are not ``texting'' via some device, reading your email, talking on your cell phone, or involving yourself in any activity that is a distraction from the instructional effort that has been prepared for you.

Electronics may be used for note taking and reference during class. However, if at any time the device is used as a distraction or prevents you from being engaged with class activities, you will be asked to cease your activity, and the expectation is that you will comply with the request. If you refuse to comply you will be asked to leave the class.

If you are called on to participate in class, the expectation is that you will be prepared to play your part in the discussion at hand. The goal of the class gatherings is to engage with the instructor, the material, and the community of your classmates.

5. Educational Objectives

This class is designed to help you learn basic data types, data structures, algorithm design, and analysis techniques including recursion. The goal is for your programming skills in an object-oriented programming language to be substantially improved. You will also become familiar with basic software engineering methodology.

Proficiency Goals:

- Object oriented approach to programming
- Dynamic memory management
- Recursive programming techniques
- Data structures, such as linked lists, stacks, queues, and trees

Familiarity Goals:

- Sorting techniques
- Searching techniques, such as binary search trees
- Algorithm run-time analysis (Big-O notation)
- User interfaces, and event-driven programs

6. Course Evaluation Questions

These questions will appear on the final course evaluation. The goal of the course is to provide a directed and comprehensive learning experience that will enable you to offer the highest rating on each of the questions. In order for that to happen, you will need to work hard to do your part – attend lectures, study the material, do the readings from the course text, complete individual exercises, programs, and labs.

37. I have confidence in my ability to solve programming problems using classes.
38. I have confidence in my ability to solve programming problems using dynamic data and pointers.
39. I have confidence in my ability to solve programming problems using recursion.
40. I understand the basic data structures (linked lists, stacks, queues, trees) and can use them in programs.
41. I understand the principles of sorting and searching.