CSci 515 Fundamentals of Programming

Course Syllabus

Spring 2015

Instructor

Derek Harter, Ph.D., Associate Professor
Department of Computer Science
Texas A&M University - Commerce
Office: SCI 355
E-mail: Derek.Harter@tamuc.edu

Class Meetings

This course will be web enhanced. Lectures, notes and course materials will be distributed through our University's eCollege online course system.

Section 03E (Lecture)	W 11:00a - 1:40p	Location: EDS 133
Section 03L (Lab)	W 2:00p - 4:00p	Location Jour 102

Course Description

This is an advanced programming course using a high level programming language, C and C++. Specific objectives are to introduce the development of algorithms as a disciplined approach to problem solving; to present programming practices in design, coding, debugging, testing and documentation of computer programs; to provide the student with the fundamental knowledge necessary for further study in the field of computational sciences.

Student Learning Outcomes:

- To understand the internal representation of the various data types.

- To review the language syntax and learn new syntax you have not previously used in programming applications.

- To correctly solve programming problems and learn how to develop algorithms.

- To examine the internal representation of two and three dimension arrays in C/C++.

- To understand dynamic memory allocation, parameter passing, the use of pointers.

Mandatory Textbook

C++ How to Program (7th Edition), by Paul J. Deitel & Harvey M. Deitel, ISBN-13: 978 0136117261

Evaluation (Tentative)

Your grade for the course will be based on the following (approximate) percentages:

First Test	20%
Second Test	20%
Labs / Assignments	30%
Final Test	30%

Final test might be in any form, a program, such as project presentation, a regular test, or a paper. The instructor will make a decision after a cooperative discussion with the students.

You should do your own work on exams/projects and for computer assignments. Copying another student's work is not acceptable. Any indication of cheating and/or plagiarism on an exam/assignment/project will be an automatic 0 (zero) for the exam/assignment/project for all students involved. Yet, based on cheating and plagiarism activity in any section of class, instructor holds the right to give F grade to the identified student(s). Regarding codes in assignments / projects, you may be required to explain the code you submitted. In case of discursive explanation, the instructor holds the right to lower your grade. You are given many programming assignments during semester. The student submitting other's source code will

take -1 for this specific assignment. A student having more than three -1 cannot gain any point for the Lab/Assignments section of class evaluation. It means that you will automatically lose 30% of your total grade.

Letter grades will be assigned according to the following scale:

Final Average (%)	Letter Grade
90 - 100	A
80 - 89	B
70 - 79	C
60 - 69	D
Below 60	F

Student's with Disabilities

The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you have a disability requiring an accommodation, please contact:

Office of Student Disability Resourcesand Services
Texas A&M University-Commerce
Gee Library, Room 132
Phone (903) 886-5150 or (903) 886-5835
Fax (903) 468-8148
StudentDisabilityServices@tamuc.edu

Academic Ethics

"All students enrolled at the University shall follow the tenets of common decency and acceptable behavior conducive to a positive learning environment." (See Student's Guide Handbook, Policies and Procedures, Conduct).

Absolutely no usage of laptops and cellular devices (texting and talking) in class. Talking and other activities that distract/disturb others in the class will not be tolerated. Instructor holds the right to ask you leave the classroom any time based on any of disturbing attitude. Each student should sign the sign-sheet if asked by instructor. Late student may not be allowed to participate the lecture.

Ethics also includes the issue of plagiarism, and copying code for programming/lab assignments is just as serious as any other type of plagiarism. If you are caught sharing or using other people's work in this class, you will receive a 0 grade and a warning on the first instance. A subsequent instance will result in receiving an F grade for the course, and possible disciplinary proceedings.

Attendance Policy

Attendance is mandatory. Students are expected to be present at all class lectures and are responsible for all material covered in class and assigned in readings. Likewise lab sessions are mandatory. Lab sessions may not be made up, and absences will affect your final course grade directly for the missed assignments.

Course Requirements and Deadlines

Credit will be given for ONLY those exams, programs, and/or projects turned in no later than the deadline as announced by the instructor of this class, unless prior arrangement has been made with the instructor. Late programs / projects / assignments can or cannot gain partial credit. Credit for late programs / projects /assignments will be announced with the description of it.

Assignments and projects will be posted in university's eCollege communication system. Detailed information will be provided by the instructor. Students also should turn in their assignment through eCollege portal. Each student is responsible for the content/instructions of email communications.

Study: To plan a minimum of three hours of outside preparation for each hour of class is a safe time allocation for successfully completing the course.

Quizzes: Unannounced pop-quizzes may be given in class and/or online through eCollege to help ensure students stay up with assigned material.

Course Schedule (Preliminary)

W	Content	Assignments
1	Introduction, Data types, Cin, Cout	Lab assg. to be completed in 20 minutes. Take-home assg: limits of data types.
2	Decision making, details of loops, string comparison, nested control structures, logical operators, == and =	Lab assg. to be completed in 20 minutes: Find averages of groups of numbers. Take-home assg.: Library inventory.
3	File Operations, text files, binary files. How to read and write, randomness. Loops to read data from file.	Lab assg. to be completed in two hours. Take-home assg: Library inventory using files
4	Functions: Modular programming, pass by reference, pass by value, ARGV details for main() function, memory details of functions.	Lab assg. to be completed in two hours. Take-home assg: Complex Math operation with functions
5	Determining the Size of a Data Type, Scope, static variables, function overloading, predefined (system) functions, default arguments	Lab assg. to be completed in two hours. Take-home assg: Student-course exercises to understand given concepts.
6	Arrays: Using with different data types, index and access to arrays. Applications to images.	Lab assg. to be completed in one hours. Take-hoem assg: Calculation of standard derivation in number arrays. Calculations in 3D arrays (TAMU-C logo)
7	Array operations, add, shift, replace, deleta an element, parallel arrays with different data types (basics of structs)	Lab assg. to be completed in two hours. Take-home assg: Frequency calculation from a text file
8	Array applications: Search in sorted and unsorted data. Sort algorithms, efficient array size calculation for a given problem	Lab assg. to be completed in two hours. Take-home assg: Sort one million number to see difference in sort algorithms.
9	Structs, memory status of structs, arrays in structs, functions with structs.	Lab assg. to be completed in two hours. Take-home assg: Calculation of relative frequency
10	Structs, structs in structs, sorting structs, sorting structs in an array	Lab assg. to be completed in two hours. Take-home assg: Vanity numbers, a departmental courseware
11	Dynamic Programming, Pointers, new and delete operators. Dynamically created arrays, dynamically created pointers.	Lab assg. to be completed in two hours. Take-home assg: Sorting Structs / Arrays using only pointer
12	Dynamic Programming with various scenarios. Friendship network, Facebook problem, Traveling salesman problem.	Lab assg. to be completed in two hours. Take-home assg: TSP