CSCE 206

Fall 2014

Instructor: **Dr. Joseph D. Hurley**
Office: Teague Bld. Room 317A, Phone 979-845-4087
Office Hours: M-W-F 10-11am, 12:30-1:15pm

Course Description: This course addresses principles of computer science including algorithm development, programming techniques, language features, and methods of documentation. Students will be introduced (in detail) to the "C++" programming language.

Required text: Starting Out With C++, Tony Gaddis.---7th ed., 2012(Pearson).

Attendance Policy: Attendance at all lectures is required to ensure that each student is abreast of the informational content of the class and becomes aware of any changes to the tentative class exam, quiz, and assignment schedules as noted later in this document. Absence from a lecture is not a valid excuse to be misinformed about any class activity and the student maintains the responsibility for all course content. **Attendance at lectures will count for 5% of your grade**. Attendance at labs will be taken but will **not factor** into the calculation of your final average.

Scholastic Dishonesty: Scholastic dishonesty will not be tolerated in any form. Working together on programming assignments can be a meaningful and real-world learning experience, BUT the final product submitted for a grade must be the work of the individual student submitting the material. Examinations are meant to measure the knowledge of individual students and offering to, or receiving any assistance from, a student that subverts this process will not be tolerated. Any identified incident of scholastic dishonesty will be dealt with severely and consequences may include a zero grade for the activity, course failure or dismissal from the university among others.

Grading Policy:

Exams:

Exam 1:	Friday: Sept. 26th.
Exam 2:	Friday: Oct. 24th.
Exam 3:	Friday: Nov. 21st.

Final:	Sections 501-503 8-8:50 lecture	Friday, Dec. 12th	10:00am-Noon.
	Sections 504-506 9:10-10 lecture	Monday, Dec. 15th	8:00am-10:00am.

ALL EXAMS ARE IN BRIGHT 124.

Only with an officially approved **university authorized absence** will any exam makeup be considered. Prior arrangements with the instructor must be made and **official verification** of circumstances necessitating the absence will be required. When calculating your semester average, **if your final exam is higher** than the lowest of your first 3 exams, the lowest will be dropped and your final used twice. If the final is the lowest of the 4 exams, it will just count for 1 of the 4 scores. Your overall exam average will constitute 70% of your semester average. Lecture attendance will be 5% of your final average.

Assignments:

Programming assignment grades will cumulatively represent 25% of your final grade. Lab assignments will be submitted on or before the due date. Lab assignment grades will lose 10% for each period late (including due date lab) up to a maximum loss of 40% (i.e., 2 weeks late). Permission to submit any assignment for grading after that period will require the direct permission of the class instructor. Assignments will require use of standard programming style conventions and documentation which will be illustrated in lab and lecture examples.

Final grades will be assigned as follows:

90-100	A
80-89	B
70-79	C
60-69	D
Below 60	F

This scale may be adjusted by the instructor to reflect score variations. It will only be potentially lowered (e.g., 78-89 for a "B"), not raised.

Lecture Topics

1.	Intro to Computers, Programming and C++	Ch 1, 2
2.	Expressions, Interactivity and Making Decisions	Ch 3, 4
3.	Loops, Files and Functions	Ch 5, 6
4.	Arrays, Searching and Sorting	Ch 7, 8
5.	Pointers, C-Strings and String Class	Ch 9, 10
6.	Structured Data	Ch 11
7.	Advanced File Operations	Ch 12
8.	Intro to Classes, Inheritance, Polymorphism	Ch 13, 14, 15

Chapter Topics

After having read and studied the following chapters you should be able to understand/explain and/or implement the following:

Chap 1: Why Program?
Computer Systems: Hardware and Software.
Programs and Programming Languages.
Input, Processing and Output.
Procedural and Object-Oriented Programming.

Chap 2: The Parts of the C++ Program.
The COUT object.
Variables, Litterals and Identifiers.
Integer and Char Data Types.
C++ String Class.
Floating-Point Data Type.
The bool Data Type.

Chap 3: The cin object.
Mathematical expressions.
Type Casting.
Assignment Statement.
Formatting Output.
Characters and String Objects.

Chap 4: Relational Operators.
If, If-Else, Nested If Statements.
Flags.
Logical Operators.
Menus.
Switch Statement.
Validating Input.

Chap 5: While, For and Do-While Loops.
Counters.
Keeping a Running Total.
Sentinels.
Software Engineering: Which Loop to Use.
Nested-Loops.
Using Files for Data Storage.

Chap 6: Software Engineering: Modular Programming.
Defining and Calling Functions.
Function Prototypes.
Passing Data by Value.
Menu Driven Programming.
Return Statement.
Local and Global Variables.
Static Local Variables.
Using Reference Variables as Parameters.
Overloading Functions.
The exit() Function.

Chap 7: Accessing Array Elements.
Bounds Checking in C++.
Array Initialization.
Processing Array Contents.
Arrays as Function Arguments.
Two-Dimensional Arrays.
Arrays with Three or More Dimensions.
Intro to STL vector.

Chap 8: Sorting and Searching Arrays.
Selection Sort Algorithm.
Bubble-Sort Algorithm.
Linear Search Algorithm.
Binary Search Algorithm.
Sorting and Searching Vectors.

Chap 9: Pointer Variables.
Getting the Address of a Variable.
Relationship between Arrays and Pointers.
Pointer Arithmetic.
Initializing Pointers.
Comparing Pointers.
Pointers as Function Parameters.
Dynamic Memory Allocation.

Chap. 10: C-Strings and String Class.
Character Testing.
Library Functions for C-Strings.
C-String/Numeric Conversion Functions.
Writing Your Own C-String Handling Functions.

Chap. 11: Abstract Data Types.
Combining Data into Structures.
Accessing Structure Members.
Initializing a Structure.
Arrays of Structures.
Structures as Function Arguments.
Returning a Structure From a Function.
Pointers to Structures.
Unions.
Enumerated Data Types.

Chap. 12: Advanced File Operations.

File Output Formatting.

Passing File Stream Objects to Functions.

More Detailed Error Testing.

Member Functions for Reading and Writing Files.

Working with Multiple Files.

Binary Files.

Creating Records with Structures.

Random-Access Files.

Opening a File for Both Input and Output.