Spring 2004

CS1410 (CS I-Introduction to Computer Science with C++) Section 20710 TTH 11:30 -1:45 pm

Instructor: Dr. Ongard Sirisaengtaksin
Office: S711 **Tel:** (713) 221-8554
Email: ongards@dt.uh.edu
Web: http://cms.dt.uh.edu/faculty/ongards/
Office Hours: MW 1:00-3:00 pm and by appointment.

Catalog description: (4-3-2) History, nature and uses of the computer-, algorithms-, number systems, information representation; and organization, with an overview of computer hardware and software, computing systems and major applications. Ethical and societal issues arc discussed. An introduction to high-level languages with an emphasis on writing programs in C++. Control statements, subprograms, data types, arrays, and an introduction to records and streams. Closed (supervised) laboratories are conducted on: an introduction to Windows operating system and a C++ programming environment; appropriate programming exercises emphasizing top-down design methodology and simple or structured data types; key topics of the discipline and areas of application-, the use of the Internet for communication and research. This course is designed as a first course for majors in Computer and Mathematical Sciences. (COSC 1420)

Course prerequisites: Credit or enrollment in MATH 1404 or MATH 1505 or MATH 1306; and placement in ENG 1301 or above.

Topic prerequisites: Knowledge and understanding of the use, terminology, and notation of mathematical functions.
Textbook: Programming and Problem Solving with C++ by N. Dale, C. Weems and M. Headington, Third edition (Jones and Bartlett).

Course grade: The course average is determined by two major tests (45%), a comprehensive final exam (30%), 5 to 7 programming projects (15%), and laboratory quizzes (10%). The course grade is determined by the standard college formula based on the course average: "A" (90-100), "B" (80-89), "C" (70-79), "D" (60-69), or "F" (0-59). Major tests are in-class, and closed book. Test dates will be announced at least one week in advance. ***There are no makeup tests and quizzes*. The student must have a passing average (60+) on both tests and programming projects to pass this course, otherwise he or she should receive at best a "D".**

Note: If you have no more six absences for the semester and your final exam score is higher than one of your test scores, your lowest test score will automatically be replaced by final exam score.

Statement on reasonable accommodations: UHD adheres to all applicable federal, state, and local laws, regulations, and guidelines with respect to providing reasonable accommodations for students with disabilities. Students with disabilities should be notified to register with Disabled Student Services and contact the instructor in a timely manner to arrange for appropriate accommodations.

Last day to drop: March 25, 2004

Final exam date: Thursday May 6, 2004 1:00 – 3:30 pm

Laboratory content:

Lab number	*Title*	*Brief description*
1	Introduction	Windows operating system; using the lab software; the Microsoft Visual C++ programming environment
2	Syntax/Expressions	Practice with C++ syntax (debugging code) and evaluating C++ arithmetic expressions
3	Data Representation	Demonstration and practice with encoding information in binary form: positive integers, characters, and simple graphics
4	Input/Output	Practice with format manipulators and file I/O
5	Control Statements	Activities involving the **if-else** statement, including Boolean expressions and nested statements
6	While-Loops	Practice with various types of **while**-loops
8	Functions I	Activities involving tracing and using predefined functions in code, as well

		as explaining and demonstrating parameter passing
9	Functions II	Practice implementing and using function prototypes based on pre- and post-conditions
10	Alternative Controls	Activities involving the **switch, for,** and **do-while** statements
11	Arrays	Diagramming, declaring, and initializing arrays; array I/O; using subscripts; character strings; passing arrays as parameters

Lecture and text content:

Chapter	*Sections*	*Topic with suggested contact time allocation (38 lecture hrs + 4 hrs testing)*
1	All	History, nature, and uses of the computer; overview of programming and problem-solving techniques; algorithms; categories of information and digital representation; converting decimal numbers to binary and vice versa; high-level vs. machine languages; the purpose of compiling, major components of a computer and a computing system; application vs. system software; the functions of an operating system (4 hrs)
2	All	Elements of C++ programs- types of tokens- C++ program structure; syntax vs. semantics; rules for identifiers; standard data types variables and constants; rules for constants; named constants; assignment statements; introduction to operators; standard stream output with the insertion operator; comments; block statements, introduction to compiler directives (5 hrs)
3	All	Arithmetic expressions- operator precedence; type coercion and casting; mixed type expressions; function calls and actual parameters; standard function libraries; value-returning functions vs. procedures; formatting output and stream manipulators (3 hrs)
4	All	Standard stream input with the extraction operator; the newline character and the *get* function; *the ignore* function; interactive I/O; file I/O; elements of top-down design; pseudocode; implementing a top-down design; documentation (3 hrs)
5	All	Boolean expressions and data; relational operators; Boolean operators; **if-else** statements; nested **if-else** statements; stream 1/0 error-checking-, pre- and post-conditions in modular design; testing control structures; tracing code (3.5 hrs)
6	All	**while**-loops; categories of loops; nested loops; major components of loops; testing loops comprehensively (3.5 hrs)
7	All	User-defined functions and modular design; void functions; actual and formal parameters; value vs. reference parameters; local variables; function prototypes vs. function definitions; function pre- and post-conditions; the *assert* function (3 hrs)
8	All	Global vs. local identifiers and scope rules; function interface design and side effects; value-returning functions; stubs and drivers (3 hrs)
9	All	**switch** statements; **do-while** loops; **for** loops; **break** and **continue** statements; loop invariants (2 hrs)
10	All	Simple data types; specialized operators; external vs. internal representation of data; user-defined data types and the **enum** keyword; named vs. anonymous data types; details of type coercion; user-written header files (2 hrs)
12	All	One-dimensional arrays; using subscripts; initializing and processing arrays; array I/O; passing arrays as function parameters; parallel arrays; non-integer subscripts (3 hrs)