Modern Programming in C++

3 credits, 3 hours
Prerequisite: MAT 1624 or equivalent course
Corequisite: ESL 1325, for English sections only
This course provides an introduction to problem solving methods and algorithm development through the study of the program, control structures, and data structures of the C++ programming language
The structure and operation of a computer; concepts and properties of an algorithm and a programming language. Introduction to procedural programming in C++, control structures, functions, recursion, arrays, painters, strings, structure, and file I/O. The main theme is that programs can be reliably designed, proven, and refined if one pays careful attention to their underlying logic, and the emphasis of the course is on the logical evolution of working programs from specifications. Students are introduced to logic-based programming methodologies which are at once powerful and practical

The grade in the class is based on three areas - each counting for one-third of the grade:
A midterm
Programming Exercises
A final Group Project

Text
Small C++, How to Program 5th Edition 2005
by Deitel and Deitel Prentice Hall
ISBN 0-13-185758-4

Office Hours:
Monday 7:30-8:00 Room B206
Tuesday 7:30-8 Room B206
9:15 to 10:15 Room B419
Wednesday 7:30-8 Room B206
Thursday 8-9 Room C352

e-mail: prof@nanamo.com
web site: http://nanamo.com

Links to free compilers may be found on the disk which came with your text.
If you use Windoze, the free Deitel document *Dive-Into Cygwin and GNU C++* clearly explains how to download and install the UNIX simulator Cygwin and the GNU C++ compiler.

MAT 1698 21511-0306 Spring 2006 J. Randall

1.Introduction to Computers and C++ Programming
- History of C and C++
- C++ Standard Library
- Typical C++ Development Environment
- Web resources

2. Introduction to C++ Programming
- Printing a line of text
- Memory Concepts
- Equality and Relational Operators

3. Introduction to Classes and Objects
- Classes, Objects, Member Functions and Data Members
- Defining a Class with a Member Function
- Defining a Member Function with a Parameter
- Data Members, *set* Functions and *get* Functions
- Initializing Object with Constructors
- Placing a Class in a Separate File for Reusability
- Separating Interface from Implementation
- Validating Data with *set* Functions

4. Control Statements: Part 1
- Control Structures
- *if* Selection Statement
- *if...else* Double Selection Statement
- *while* Repetition Statement
- Counter-Controlled Repetition
- Sentinel-Controlled Repetition
- Nested Control Statements
- Assignment Operators
- Increment and Decrement Operators

5. Control Statements: Part 2
- Counter-Controlled Repetition
- *for* Repetition Statement
- *do...while* Repetition Statement
- *switch* Multiple-Selection Statement
- *break* and *continue* Statements
- Logical Operators
- Equality (==) and Assignment (=) Operators

6. Functions
- Math Library Functions
- Function Definitions with Multiple Parameters
- Function Prototypes and Argument Coercion
- C++ Standard Library Header Files
- Storage Classes
- Scope Rules
- Function Call Stack and Activation Records
- Functions with Empty Parameter Lists
- Inline Functions
- References and Reference Parameters
- Default Arguments
- Unary Scope Resolution Operator
- Function Overloading
- Function Templates
- Recursion
- Recursion versus Iteration

7. Arrays and Vectors
- Declaring Arrays
- Passing Arrays to Functions
- Searching Arrays with Linear Search
- Sorting Arrays with Insertion Sort
- Multidimensional Arrays
- C++ Standard Library Class Template *vector*

8. Pointers and Strings
- Pointer Variable Declarations and Initialization
- Pointer Operators
- Passing Arguments to Functions by Reference
- Using *const* with Pointers
- Selection Sort using Pass-by-Reference
- *sizeof* Operators
- Pointer Expressions and Pointer Arithmetic
- Relationship Between Pointers and Arrays
- Arrays of Pointers
- Function Pointers
- Pointer-Based String Processing