Las Positas College
3000 Campus Hill Drive
Livermore, CA 94551-7650
(925) 424-1000
(925) 443-0742 (Fax)

Course Outline for CS 1

COMPUTING FUNDAMENTALS I

Effective: Fall 2010

I. CATALOG DESCRIPTION:
CS 1 — COMPUTING FUNDAMENTALS I — 4.00 units

Introduction to programming and problem-solving using C++. Problem solving techniques and algorithms; program design, development, style, testing and debugging. C++ syntax covered includes: variables; data types; operators and expressions; control structures; library and user-defined functions; basic input/output; arrays; user-defined data structures.

3.00 Units Lecture 1.00 Units Lab

Strongly Recommended
MATH 107 - Pre-Algebra

Grading Methods:
Letter or P/NP

Discipline:

	MIN
Lecture Hours:	54.00
Lab Hours:	54.00
Total Hours:	108.00

II. NUMBER OF TIMES COURSE MAY BE TAKEN FOR CREDIT: 1

III. PREREQUISITE AND/OR ADVISORY SKILLS:

Before entering this course, it is strongly recommended that the student should be able to:

A. MATH107

IV. MEASURABLE OBJECTIVES:

Upon completion of this course, the student should be able to:

A. write, edit, compile, run and debug C++ programs;
B. carry out program development steps;
C. design and implement algorithms as C++ programs;
D. explain and apply C++ control structures for sequencing, selection and iteration;
E. explain and implement user-written functions in C++;
F. create and interpret expressions involving arithmetic and logical operators;
G. use elementary input and output operations;
H. use C++ to solve problems involving one-dimensional arrays and simple user-defined data structures;
I. produce functional and user-friendly programs of short to medium length.

V. CONTENT:

A. Tools and Procedures
    1. Creation, editing and building programs
    2. Testing and debugging programs
B. Problem solving concepts
    1. Problem definition and algorithm specification
    2. Documentation standards
    3. Use of structured programming elements: sequence,
    4. iteration, selection
    5. Testing and debugging methods.
C. Elementary C++ Syntax Elements
    1. Built-in data types
    2. Literals
    3. Numeric and string constants
D. C++ Expressions and Assignment Statements
    1. Operators: arithmetic, logical, relational
    2. Precedence and type rules
    3. Assignment statements

        4. Libraries
    E. C++ Control structures
        1. IF and IF-ELSE statements
        2. SWITCH statements
        3. FOR, WHILE and DO-WHILE statements
        4. Nesting of control structures
    F. C++ Elementary Input/Coutput
        1. cin and cout
        2. input/output of numeric, character and string data
        3. elementary reading/writing of text files
    G. Modular Programming and C++ Functions
        1. Using existing functions
        2. Function declaration and calling
        3. Parameter passing - by-value and by-reference
        4. Void vs value-returning functions
    H. One-Dimensional C++ Arrays
        1. Declaring and accessing one-dimensional arrays
        2. Arrays as function parameters
        3. Searching within an array
        4. Typical array uses (e.g., summation)
    I. C++ Strings
        1. Operators: concatenation, character access
        2. C++ string functions
        3. Comparison with C-style strings
    J. C++ Structs
        1. Declaring new struct types
        2. Accessing struct components
        3. Structs as function parameters

VI. METHODS OF INSTRUCTION:

A. **Lecture** -

VII. TYPICAL ASSIGNMENTS:

A. Programming
    1. Create an interactive C++ program that inputs, displays and finds the mean of an array of floats.
    2. Create a C++ program that reads a file of words, finds and reports the number of words longer than given length.
B. Reading
    1. Read the section of Chapter 3 on while loops; Try and check the practice exercises for that section.
    2. Read the on-line documentation on functions in the math library; Write an example of two different functions being used in a program.

VIII. EVALUATION:

A. **Methods**

B. **Frequency**

1. At least two in-class midterm examinations, or one in-class midterm examination and several quizzes
2. One in-class comprehensive final examination
3. Programming assignments to cover each topic within the course content (contents can be combined). Approximately one programming assignment per week is recommended.
4. Recommended Weighting
    a. In-class activities (including exams and quizzes): No less than 65%
    b. Other: No more than 35%

IX. TYPICAL TEXTS:

1. Savitch, W *Problem Solving with C++* . 3rd ed., Addison Wesley, 2008.
2. Gaddis, T. *Starting Out with C++: From Control Structures through Objects* . 6th ed., Addison Wesley, 2008.

X. OTHER MATERIALS REQUIRED OF STUDENTS:

A. A portable storage device (e.g, USB drive) is strongly recommended.