

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

Course Outline for CS 30

C++ PROGRAMMING

Effective: Spring 2017

I. CATALOG DESCRIPTION:

CS 30 — C++ PROGRAMMING — 4.00 units

Applications programming in C++ for the student already familiar with the concepts of programming. The following concepts are introduced: Object-oriented programming, encapsulation, inheritance and polymorphism; introduction to data abstraction and structures; pointers; file I/O.

3.00 Units Lecture 1.00 Units Lab

Prerequisite

CS 1 - Computing Fundamentals I
with a minimum grade of C

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 the course a student should be able to:

A. CS1

IV. MEASURABLE OBJECTIVES:

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

- A. GENERIC: These outcomes are being developed throughout the entire programming sequence. Upon completion of the course, to an intermediate level, students should be able to: Programming Skills
 1. Present the elements and features of the development environment
 2. Explain and use the design process
 3. Define and use functions and storage classes
 4. Define and explain trends in programming standards
 5. Write, compile, test and debug programs
 6. Present the characteristics of object-oriented programming
 7. Define and use data types and variables
 8. Define and use multi-dimensional arrays
 9. Define and use constructor and destructor functions
 10. Define and use function overloading
 11. Define and use operator overloading
 12. Define and use inheritance mechanisms in OOP
 13. Define and use user interfaces
 14. Define and use file I/O
 15. Define and develop class modules
 16. Develop and use event-driven programs
 17. Use pointers to data, objects and functions
 18. Use dynamic memory allocation
- B. Systems Analysis
 1. Develop high-level systems and functional specifications
 2. Define general scope of work to meet requirements and constraints
- C. Systems Design
 1. Specify major subsystems and interfaces
 2. Develop detail design specifications
 3. Select design methodology and tools
 4. Identify maintenance requirements

- D. Technical Documentation
 - 1. Write in a concise and precise form appropriate for technical documentation
 - 2. Explain and use the processes and techniques of technical documentation
 - 3. Record system specifications accurately and completely
- E. Testing and Debugging
 - 1. Select debugging and testing methodology, and develop comprehensive and systematic test plan
 - 2. Develop testing procedures
 - 3. Conduct tests in the most efficient way
 - 4. Test programs, and document errors and solutions
- F. User Interface Design
 - 1. Define the requirements for the user interface
 - 2. Detail the development process and methods best suited for the project
 - 3. Develop user interface (UI) to meet user requirements
 - 4. Test UIs
- G. Problem Solving
 - 1. Recognize a wide range of problems, and assess their impact on the system
 - 2. Use a wide range of troubleshooting methods and tools to isolate problems
 - 3. Select the appropriate approach to identify causes of the problem based on the given situation
- H. SPECIFIC: These outcomes are detailed specifically for this course. Upon completion of the course students should be able to: Give an overview of the evolution and present state of programming languages
- I. Describe and employ basic principles of software engineering.
- J. Define and use abstract data types in program applications.
- K. Define and employ overloading of functions and operators.
- L. Write functions implementing iteration.
- M. Define and illustrate encapsulation, inheritance and polymorphism in C++.
- N. Write programs that use file I/O techniques.
- O. Write programs as console applications and windows applications.

V. CONTENT:

- A. Review of Fundamental C++
- B. Classes & Objects
 - 1. Data member
 - 2. Constructors
 - 3. Destructors
 - 4. Member Functions
 - 5. Member Operators
 - 6. Public, Private, Protected
- C. Database
 - 1. Create
 - 2. Modify
 - 3. Delete
- D. UI & Prototyping
- E. Pointers & Dynamic Allocation
- F. Inheritance
 - 1. Single
 - 2. Multiple
- G. File I/O
 - 1. Stream
 - 2. System
 - 3. Standard
- H. Event vs. Sequential Programming

VI. METHODS OF INSTRUCTION:

- A. **Lecture** -
- B. **Demonstration** -
- C. **Projects** - Optional: Programming project completed in teams
- D. **Lab** - Lab Programming Assignments
- E. **Discussion** -

VII. TYPICAL ASSIGNMENTS:

A. Write a program that asks a workers age and years of service and determines whether that worker is eligible for retirement based on the following rule: 1. All employs are eligible to retire at 65. 2. All employees are eligible to retire after 35 years service. 3. At 60, employees may retire with 25 years service. 4. At 55, employees may retire with 30 years service. B. Write a program that reads a list of names and bowling scores from a text data file and produces a report. The report will be displayed on the screen and printed to a text output file for printing. C. Write a GUI application that inputs data from edit boxes and stores the information in a class object. Have the application also write the information to a binary file.

VIII. EVALUATION:

- A. **Methods**
- B. **Frequency**
 - 1. Frequency of evaluation
 - a. Recommend 2 or 3 exams plus final examination
 - b. Recommend programming assignment to cover each topic within course content. Contents can be combined.
 - 2. Types of Exam Questions
 - a. Write the C++ command to open a file in binary, read only, mode using system I/O.
 - b. Write the C++ command to allocate 100 bytes from the heap and assign the address to pointer variable p.
 - c. Write the C++ command to deallocate an array of 10 integers called list.

IX. TYPICAL TEXTS:

- 1. Hubbard, John *Programming with C++*, ., McGraw-Hill (Schaum's Outline), 1996.
- 2. Prata, Stephen *C++ Primer Plus*. 3rd ed., Sams Publishing (The Waite Group), 1998.
- 3. Schildt, Herbert *C++: The Complete Reference*. 3rd ed., Osborne McGraw-Hill, 1998.

X. OTHER MATERIALS REQUIRED OF STUDENTS:

- A. C++ compiler is necessary
- B. Current version of Microsoft Visual C++
- C. Current version of Borland's C++, Unix or Linux C++ compilers.