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

Course Outline for CS 36

WINDOWS AND MFC PROGRAMMING

Effective: Fall

I. CATALOG DESCRIPTION:
CS 36 — WINDOWS AND MFC PROGRAMMING — 4.00 units

This is an advanced course in Windows programming using C++. Teaches Applied Windows Programming in C++. This course presents a comprehensive introduction to the Windows C++ programming and its role in the Internet and database programming. A variety of OOP topics covered will include building basic Windows applications including menus, dialog boxes, main window, buttons, MFC Wizards, ODBC, OLE-DB/ADO, DHTML, and ActiveX.

3.00 Units Lecture 1.00 Units Lab

Prerequisite
CS 30 - C++ Programming
or

CS 2 - Computing Fundamentals II
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. CS30
1. 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
2. Present the elements and features of the development environment
3. Explain and use the design process
4. Write, compile, test and debug programs
5. Present the characteristics of object-oriented programming
6. Define and use data types and variables
7. Define and use user interfaces
8. Systems Design
9. Specify major subsystems and interfaces
10. Develop detail design specifications
11. Technical Documentation
12. Write in a concise and precise form appropriate for technical documentation
13. Testing and Debugging
14. Test programs, and document errors and solutions
15. User Interface Design
16. Define the requirements for the user interface
17. Problem Solving
18. Recognize a wide range of problems, and assess their impact on the system
19. Use a wide range of troubleshooting methods and tools to isolate problems
20. Select the appropriate approach to identify causes of the problem based on the given situation
21. 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
22. Describe and employ basic principles of software engineering.
23. Define and use abstract data types in program applications.
24. Define and employ overloading of functions and operators.
25. Write functions implementing iteration.
26. Define and illustrate encapsulation, inheritance and polymorphism in C++.

27. Write programs that use file I/O techniques.
28. Write programs as console applications and windows applications.

B. CS2

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 advanced level, students should be able to: Programming Skills
   1. Explain and use the design process
   2. Define and use functions and storage classes
   3. Define and use dynamic data structures
   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 pointers
   8. Define and use constructor and destructor functions
   9. Define and use function overloading
   10. Define and use operator overloading
   11. Define and use inheritance mechanisms in OOP
   12. Define and use user interfaces
   13. Define and use file I/O
   14. Define and develop class modules
   15. Develop and use event-driven programs
   16. Embed one language in another

B. Database Design
   1. Explain database design concepts and the role of database components
   2. Create and customize forms and reports
   3. Explain the use of databases and information in the business environment
   4. Develop database business applications

C. Testing and Debugging
   1. Select debugging and testing methodology, and develop comprehensive and systematic test plan
   2. Design testing programs to uncover hardware compatibility problems during the development phase of the project
   3. Develop testing procedures
   4. Conduct tests in the most efficient way
   5. Test programs, and document errors and solutions

D. User Interface Design
   1. Define the requirements for the user interface
   2. Define candidate solutions to business problem, and select best approach with client
   3. Detail the development process and methods best suited for the project
   4. Develop user interface schema to meet user requirements
   5. Develop and test prototypes
   6. Make recommendation for design changes based on prototyping test results
   7. Participate and conduct design and development reviews
   8. Document design and development, and changes in design according to company policies
   9. Construct user interfaces for flexibility and adaptability
   10. Perform user interface tests, and troubleshoot problems
   11. Follow organization and industry standards for development
   12. Train system user on interface and perform user validation

E. 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

F. Task Management
   1. Break down projects and activities into a series of tasks

G. SPECIFIC: These outcomes are detailed specifically for this course. Upon completion of the course students should be able to: Write programs using Single Document Interface.

H. Write programs using Multiple Document Interface.

I. Write programs using Visual and Nonvisual Components

J. Write programs using Classes.

K. Create Help files

L. Write programs using Windows API.

M. Write programs using MFC.

N. Demonstrate how to packaging a windows application

O. Use the windows Debugger.

V. CONTENT:

A. Single Document Interface Programming

B. Multiple Document Interface Programming

C. Visual Components

D. Nonvisual Components

E. Classes
   1. Heirarchy
   2. Inheritance

F. Help files and resources

G. Prototyping

H. Windows API

I. MFC

J. Packaging

K. Debugging

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 Windows GUI application to place a user designed graphic on the screen. 1. An example might be to have the students design a television using the Shapes Components 2. Apply buttons, a mainmenu, or other appropriate components to the television to simulate use. B. Write a GUI program that asks a workers age and years of service and determines whether that worker is eligible for retirement based on the following rules: 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. 5. Input comes from data components: checkboxes, radiogroups, editboxes, etc. C. An optional team project to write a program to create a WEB browser with an attached database which tracks users, passwords, and web sites visited.

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. What component would you use to cause a label to blink?
   b. Describe the difference between a visual and non-visual component.
   c. What API call would you use to copy files from one device to another (open book exam).
   d. Describe the difference between an SDI and MDI application. Give an example of each.

IX. TYPICAL TEXTS:

1. - *Desktop Applications for Microsoft VC++ 6.0: MCSD Training Kit.*, Microsoft Press, 1999.
2. Gary J. Bronson *A First Book of Visual C++.*, Brooks/Cole Pub. Co., 1999.
3. Jeff Prosise *Programming Windows With MFC.* 2nd ed., Micrsoft Press, 1999.
4. Harvey, Dr. Deitel, Harvey M. Deitel, Paul J. Dietel, Edward T. Strassberger *Getting Started with Visual C++ 6 with an Introduction to MFC.* , Prentice Hall, 1999.

X. OTHER MATERIALS REQUIRED OF STUDENTS:

A. Current version of Microsoft Visual C++
B. Current version of Borland’s C++
C. Current version Linux C++ compilers