

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

Course Outline for CS 32
VISUAL BASIC PROGRAMMING
Effective: Fall 2003

I. CATALOG DESCRIPTION:

CS 32 — VISUAL BASIC PROGRAMMING — 4.00 units

Applications programming in Visual Basic to create Windows oriented applications for student already familiar with the concepts of programming. Emphasis on algorithms, I/O, multimedia capabilities, data file I/O, loops, decision-making, string processing, functions, control objects (such as sliders, combination boxes, radio buttons), ActiveX controls, Object Linking and Embedding (OLE), use of the grammar and syntax for a foundation for VBScript and Visual Basic for Applications.

3.00 Units Lecture 1.00 Units Lab

Prerequisite

CS 1 - Computing Fundamentals I
or

-

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;
 - 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 user interfaces;
 - 10. define and use file I/O;
 - 11. define and develop class modules;
 - 12. develop and use event-driven programs;
- 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: write programs using the basic grammar and syntax of Visual Basic.
 - I. write programs using OOP concepts.
 - J. write programs using VBScript in web applications.
 - K. write programs using VBA and OLE.
 - L. write programs using advanced language capabilities: ActiveX, multimedia, timer, mouse controls, and graphical images.
 - M. write programs using specialized capabilities (such as User Defined structures, SQL and Multiple Document Interface structures).

V. CONTENT:

- A. Introduction to Object Oriented Programming (OOP) concepts
- B. Use of Visual Basic as a rapid applications development (RAD) tool
- C. VBScript in web applications programming
- D. Visual Basic for Applications (VBA) to enhance Microsoft's Office Suite products such as Access
- E. Language grammar and syntax:
 1. loops
 2. if conditions
 3. I/O capabilities
 4. file processing (sequential and random files)
 5. variables
 6. string processing
 7. display format
 8. arrays
 9. case structure
- F. Advanced language capabilities
 1. ActiveX tools
 2. multimedia
 3. timer
 4. mouse controls
 5. graphical images
- G. OLE (Object Linking and Embedding) capabilities to Microsoft's Office suite products (Excel, Word and Access).
- H. Specialized capabilities (such as User Defined structures, SQL and Multiple Document Interface structures).

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. Create your own online Visual Basic restaurant. 1. The customer can order one of several different dishes: a. Main Entrée (minimum 6 choices) b. Side Order 1 (minimum 5 choices) c. Side Order 2 (minimum 5 choices) d. Dessert (minimum 4 choices) e. Drink (minimum 5 choices) 2. Your logic will compute each individual dish, total what they ordered, charge sales tax (8.5%) and show the grand total. You decide on a family "theme" restaurant type (i.e. Cajun, Steak, Chinese, Vietnamese, French, Mexican, Seafood, etc.). Decide on each item's prices (keep it somewhat "believable"). For example: a. Main Entrée - Kansas City Porterhouse Steak, Chopped Steak, Hamburger with Mushroom Gravy, T Bone Steak, Teriyaki Sirloin Steak b. Side Order 1 - buttermilk Mashed Potatoes, Potatoes Au Gratin, Baked Potato, French Fries, Rice Pilaf c. Side Order 2 - Steamed Vegetables, Chef Salad, Soup of the Day, Sautéed Vegetables, Pasta with Meat Sauce d. Dessert - Chocolate Cake, Chocolate Torte, Mixed Fruit Pie, Vanilla Pudding e. Drink - Coffee Latte, Decaf Coffee, Ice Tea, Hot Tea, Milk 3. You'd probably want to use a Combo Box for each entrée type (that means a Form Load with Additents). If they change their mind, the displayed prices should also change (picture box?). Control is probably in the Combo Boxes and in one "Calculate" button. 4. Include a NONE in each combo box choice (the customer may only want to buy one thing a la carte). If they buy "nothing", a MSGBOX should pop up telling them...."Order Something of Quit" (I leave the message content up to you. 5. Include two "specials" using "radio" buttons. Push the button and the special appears on the form. For example: a. Blackened Red Fish b. Shrimp Creole c. Steamed Vegetables d. Garlic Bread e. Coffee f. Soda 6. I leave the pricing up to you. The "Aloha Plate" example I did in class is a model for this last assignment, have "fun". 7. Using a Timer, add a "Splash" screen that appears initially and then disappears to be replaced by the Order form: a. form1.unload b. form2.show 8. The Timer is set in milliseconds. B. 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. C. 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

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 a function to prompt the user for a choice of foods and have the function return the caloric value of the choice:
 - 1. Fish = 100
 - 2. Meat = 200
 - 3. Bread = 50
 - 4. Other = 0
 - b. Use a loop structure to add the numbers from 1 to 100.
 - c. Write a function to read in a set of integers and calculate the average of the numbers.
 - d. Write the line of code to display a formatted line as follows: Counter(2) _ _ Name (20) _ _ SSN(11) _ _ Age(3) _ _ \$Salary (6.2) (use the variable names as indicated and with the widths of the fields indicated in the parentheses)

IX. TYPICAL TEXTS:

- 1. Julia Bradley and Anita Millsbaugh *Programming in Visual Basic 6.0.*, Irwin-McGraw-Hill, 1999.
- 2. Edward J. Coburn *Programming with Visual Basic 6.0.*, Brooks/Cole, 2000.
- 3. Kip Irvine & Kaiyang Liang *Advanced Visual Basic 6* . 2nd ed., Scott Jones Publishing, 1999.

X. OTHER MATERIALS REQUIRED OF STUDENTS:

- A. Computer lab student machines running the latest Microsoft Windows operating system with the latest version of Microsoft Visual Basic