

Course Description

Through a combination of lectures, demonstrations, lab activities, and project assignments, this course will cover the basic concepts and techniques of algorithm development and programming. Visual Basic will be used as the programming language. The course is the first of a series of core courses required for a major or a minor in *Computer Information Systems* at Chaminade University. Although there are no specific programming prerequisites for the course, a familiarity with the use of the microcomputer under the Windows environment is assumed.

Course Objectives

The course is intended to help the students to:

- Become acquainted with the basic concepts of the computer, its organization, and software systems
- Become familiar with the top-down, stepwise refinement approach to algorithm design
- Become familiar with the modular approach to program development
- Learn a collection of basic algorithms
- Become acquainted with event-driven programs
- Learn the basic features of the Visual Basic language
- Develop basic problem-solving skills using the computer
- Lay a basis for subsequent courses in Computer Information Systems

Text Book

Essentials of Visual Basic 6.0 Programming, by David Schneider.
Prentice Hall, 1999. ISBN 0-13-012720-5

[Return to Top](#)

Topics

Topics covered in this course will include the following:

- Distinction among *algorithm*, *pseudocode*, and *program*
- Steps in Programming
- VB Basics
 - Objects and events
 - Event procedures
 - Variables
 - Data types
 - Operators
 - Built-in functions
 - Character Strings
- Input and Output
 - Reading from File
 - Input Box
 - Outputting to Printer
- General Procedures
 - Sub procedures
 - Procedures with arguments
- User-defined Functions
- Decision Structures
- Repetition Structures
 - Do-while loop
 - For-next loop
- Arrays
- Form-level Declarations
- Algorithms Involving Arrays
 - Maximum, Minimum
 - Searching
 - Sorting

[Return to Top](#)

Requirements

The following is a summary of what is expected of you for the course. Refer to the section on [Grading](#) for further details.

- Class Attendance
- Readings and Exercises
- Project Assignments
- Lab Exercises
- Tests
- Final Exam

Class attendance is important, since main points of the course will be highlighted and details and examples will be demonstrated in the

class. Readings will provide you with further explanations on the concepts and techniques covered in the class. Exercises will help you to reinforce key ideas covered in the class and to prepare you for tests. (Many of the test questions will be based on such Exercise questions). Since you learn by doing, *project assignments* will be the most important element among your responsibilities in the course.

[Return to Top](#)

Project Assignments (PAs)

Ten project assignments are scheduled for the semester. The schedule for project assignments are indicated in the class [Schedule](#). Generally speaking, you will have several days to finish your PAs. Get in the habit of starting early on your assignments. Late assignments will be assessed a penalty. Regular credits will not be given for a program that is past its due date by one week or more. (However, a minimum value will be entered in the database for assignments that were turned in very late to distinguish them from those that were not turned in at all.) The procedure for submitting assignments will be announced in the class.

If you have difficulty with your programs, get help immediately so that you can stay on top of your assignments. When you are seeking help from the instructor in debugging your programs, always accompany your questions with a hardcopy of your program listing or a copy of your algorithm written in pseudocode.

You are expected to spend extra time beyond the class time on your project assignments. Check for posted computer lab hours in H124. Be conscientious in completing your assignments, since they are indispensable to learning algorithm development and programming.

[Return to Top](#)

Tests

Four tests, in addition to the final exam, are scheduled for the semester. Their dates are indicated in the course [Schedule](#). These tests will cover concepts and programs covered in the class and in project assignments. In general, there will be no make-up quizzes and tests. Special cases will be considered when there are valid reasons, but arrangements must be made *before* the scheduled test dates.

[Return to Top](#)

Submitting One's Own Work

Each student is expected to write his or her own programs that are to be submitted as project assignments. Although modern programming practices require extensive teamwork, one of the main goals in this class is that each student learn the basic programming skills by practicing individually. You must distinguish between consulting your friends or discussing problems with them from copying other people's work. Even if you "work together," each submission must be your own and be different from a work by another student. The penalty for copying in tests, quizzes, and project assignments is, for the first offense, a grade of 0 for all parties involved; for the second offense, an *F* for the course.

Attendance

Regular class attendance is important since you are responsible for all materials covered in the class. Attendance will be taken at all class sessions, and it will be considered as part of your final grade. If you need to be absent from a class, you should let the instructor know so that he can help you in catching up. Generally speaking, there will be no make-up tests. Make-up tests will be considered, with prior arrangements, only for excused absences because of serious reasons. Be sure to inform the instructor when you foresee that you must miss a test. A missed test due to an unexcused absence receives a score of 0.

[Return to Top](#)

Grading

The semester grade will be based on the following elements of your course responsibilities.

Tests:	60 x (4) =	240
Project Assignments:	25 x (10) =	250*
Lab:	10 x (10) =	100**
Class Participation (attendance):		50
Final Exam:		100

Total:		740**

* A minimum of 7 completed assignments is a necessary condition for a passing grade.

** The total is subject to change depending on the number of quizzes.

The following guidelines will be used in determining the final grades.

A: ≥ 90 ; B: ≥ 80 ; C: ≥ 70 ; D: ≥ 60 ; F: < 60

Test dates are indicated in the Course Schedule.

[Return to Top](#)

Criteria for Project Assignments

Generally speaking, for each programming assignment submit:

- hardcopy of the source code
- image of the program interface

An exception is PA No. 1, which requires only the interface. How to produce the hard copy and an image of the interface will be explained in the class. If there are more than one problem involved, submit a hardcopy and a screen image for each program. The grade for each assignment will be based on the correctness and completeness of both the interface and the code.

Your projects will be evaluated on the following points.

- Correctness--does it work as advertised?
- Completeness--does it satisfy all requirements in the problem description?
- Understandability--is the code easy to understand and is written in the simplest way possible?
- Interface--is the interface orderly, attractive, and easy to understand?

Check the following points in your code.

- Include documentation--a) name, b) PA number, and c) date.
- Include summary description of program.
- Insert the *Option Explicit* statement (declare all variables).
- Indent the body of subprograms and functions.
- Indent the body of decision and loop structures.
- Insert a comment before a major section in the code and a blank line after it.
- Insert a space before and after each operator; e.g., $c = a + b$,
not $c=a+b$.
- Begin each variable name in lowercase; each procedure name

in uppercase.

- Each function and subprogram should be logically simple; e.g., a procedure that inputs data should not also calculate.
- All interface objects should have user-defined names instead of default names--e.g., txtName.

[Return to Top](#)

Getting Help

For "quick" questions an [email](#) is the simplest way to contact me. Feel free to drop in at my office (Kieffer Hall, 14) during office hours or to set up an appointments outside those hours. (I am around my office usually in the afternoon.) When you need help in debugging programs, bring a copy of the code that is causing the problem.

[Return to Top](#)



[Syllabus](#)

[Schedule](#)

[Resources](#)

[Records](#)

[Office
Hrs](#)

[Home](#)

Wk	Date	Topics	Readings	Exercises	Lab	Due
1	8/26	I. Introduction ● Preliminaries ○ Pretest ○ Syllabus ○ Terminology ○ Local System ○ Printing screen dump (for PAs)	● 1.4 Intro to VB (p22-25) ● 15. Programming Tools (p25-26)			
	8/28				Lab 1	
	8/30	● Problem Solving ○ VB Problem Solving ○ Algorithms ○ Pseudocode				
2	9/2	Martin Luther Day (No Classes)				
	9/4	II. VB Programming Basics ● Objects & Events ○ Objects ○ Events Procedures ● Number Type ○ Arithmetic Operations ○ Variables ○ Relational Operators ○ Built-in Functions	● 2.1 Visual Basic Objects (p36-47) ● 2.2 VB Events (p50-58) ● 2.3 Numbers (p63-69)	Practice Problems 2.1: ● 1, 2	Lab 2	
	9/6			Exercises 2.1: ● 1,3,5 ● 30, 31, 37 Exercises 2.2: ● 1,3,5,7 ● 37, 39, 41		● PA 1
3	9/9	● String Type ○ Literals & Variables ○ Concatenation ○ Declaring Strings ○ Scope of Variables ○ Text Box for I/O	● 2.4 Strings (p 77-84)	Practice Problems 2.3: ● 1, 2		
	9/11			Exercises 2.3: ● 1,3,5 ● 23,25,27 ● 35,39	Lab 3	
	9/13					

		○ ANSI (ASCII) Character Set			● PA 2
4	9/16	○ String Relationships ○ String Functions	● String Relationships (p84) ● String Functions (p85-89)	Practice. Problems 2.4: ● 1,2,3 Exercises 2.4:	
	9/18			● 1,3,5,7 ● 17,19,21,23 ● 53,55	Lab 4
	9/20	Test 1 <u>Review Questions</u>			● PA 3 ● Example
5	9/23	● Input and Output ● Reading from File ● App.Path method ● Input from Input Box	● 2.5 Input & Output (p96-103) ● Message Box (p103)	Practice. Problems. 2.5: ● 1, 2	
	9/25	● Continuation Character ● Comment Statements ● Output to Printer	● Continuation Char (p104)	Exercises 2.5: ● 1,3,5,7 ● 15,17,19,21 ● 29,31	Lab 5
	9/27		● Output to Printer (p104-107)		
6	9/30	● General Procedures	● 2.6 general Procedures (p115-120)	Practice. Problems. 2.6:	
	10/2	● Sub Procedures ● Procedures with Arguments	● Variables & Expressions As Arguments (p 120-122) ● Passing values Back From Sub Procedures (p122-125)	● 1 Exercises 2.6: ● 1,3,5,7 ● 27,29	Lab 6
	10/4				● PA 4
					● Extra 1
7	10/7	● Passing Values Back ● Function Procedures	● Function Procedures (p125-130)	Practice. Problems. 2.6:	
	10/9	○ Built-in Functions ○ C to F Conversion ○ lb to kg		● 2 Exercises 2.6: ● 33,35,3	Lab 7
					● PA 5

		Conversion ○ Hypotenuse of Rt Triangle1				
	10/11	Test 2 (Review Guide)				
8	10/14	Discoverer's Day (No classes)				
	10/16	III. Controlling Program Flow	• 3.1 Decision Structures (p146-148) • Logical Operators (p152-155)	Practice. Problems. 3.1: • 1, 3 Exercises 3.1: • 1,3,5 • 11,13,15 • 21,23 • 29,31,33	Lab 8	
	10/18	• Decision Structure ○ If-Else Structure ○ Logical Operators ○ Compound Conditions ○ If-Else-If Structure				
9	10/21	• Do While Loop • Algorithms with Do While Loops	• 3.2 Do Loop (p164-166)	Exercises 3.2 • 1,3,5 • 7,9 • 17,19,21		• PA6 • Extra 2
	10/23				Lab 9	
	10/25					
10	10/28	• Returning to File I/O	• Input & Output (p96-100)			
	10/30				Lab 10	• PA7
	11/1	Test 3				
11	11/4	• For...Next Loop IV. Arrays • Declaring Arrays • Form-level Declaration • Form-Load Event • Parallel Arrays	• For...Next Loop (p177-183) • 4.1 Creating & Accessing Arrays (p200-209)	Practice. Problems. 3.3: • 1,2 Exercises 3.3: • 1,3,5,7 • 15 • 17,19,21 Practice Problems 4.1: • 1,2		
	11/6				Lab 11	• Extra 3
12	11/11	• Array Algorithms • Sorting • Searching	• 4.2 Sorting & Searching (p217-221) • Searching	Exercises 4.1: • 1,3,5 • 7,9		

	11/13		(p222-225)	19,21,23,25 Practice Problems 4.2:	Lab 12	PA8
	11/15			1,2 Exercises 4.2: 1,3,5,7 17,19,21		
13	11/8	Binary Search				Extra 4
	11/20				Lab 13	
	11/22	Test 4				
14	11/25	Arrays & Sequential Files	4.3 Arrays & Sequential Files (p231-237)	1 Practice Problems 4.3: Exercises 4.3: 1,3 5 7,9,11,13 17		
	11/27				Lab 14	PA9
	11/29	Thanksgiving Recess (No Classes)				
15	12/2	- Problems involving Sort and Search Algorithms - Review & Catching Up				
	12/4					
	12/6					- PA10 - Extra 5
16	12/9	Final Exam: (10:30-12:30)				