manyama

R Mam ma ? FO OU

NONAME

CIS 150: Introduction -16-Programming

Syllabus











Syllabus Schedule 5 chedule 5 chedu

Resources

Office Hours

Record Check

Questionnaire

150 Home Home

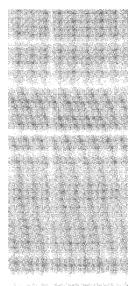
Course Description

Through a combination of lectures, lab activities, and project assignments, this course will cover the basic concepts and techniques of programming and algorithm development. 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

Information Systems

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

Text Book

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

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
- . Quizzes (unscheduled)
- . Tests
- Final Exam

Class attendance is important, since main points of the course will be highlighted and details will be demonstrated in the class. Readings will show you background and 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

Ten project assignments are scheduled for the semester. In general, you will have one week to complete an assignment. The schedule for project assignments are indicated in the class Schedule. Get in the habit of starting early on your assignments. Late assignments will be assessed a penalty which is equivalent to one letter grade per day. 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.

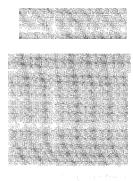
If you have difficulty with your programs, get help immediately so that you can stay on top of your assignments. When you are seeking help from the instructor in debugging your programs, always accompany your questions with a hardcopy of your program listing or a copy of your algorithm written m pseudocode.

You are expected to spend extra time beyond the class time on your project assignments. Check for posted computer lab hours. 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 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 quiz or test dates.

Return to Top

Submitting One's Own Work

Each student is expected to write his or her own programs. 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 work must be different from another. 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. Attendance will be considered as part of your final grade. If you need to 6 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 grade of 0.

Return to Top

Grading

The semester grade will be based on the following elements of your course responsibilities. (Total is subject to change depending on the number of quizzes.)

Tests: $60 \times (4) = 240$ Project Assignments: $25 \times (10) = 250$

```
Quizzes 5 x (10) = 50
Class Participation (attendance): 50
Final Exam: 100
Total! 690
```

* A minimum of 7 completed assignments is a necessary condition for a passing grade. 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. l, which requires only the interface. How to produce the hard copy and an image of the interface will be discussed in the class. (You can also refer to the How To section.) 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?
- 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-name, PA ID, etc.
- . Include summary description of program.
- . Insert the *Option Explicit* statement (declare all variables).
- . Indent the body of subprograms and functions.
- . Indent the body f 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

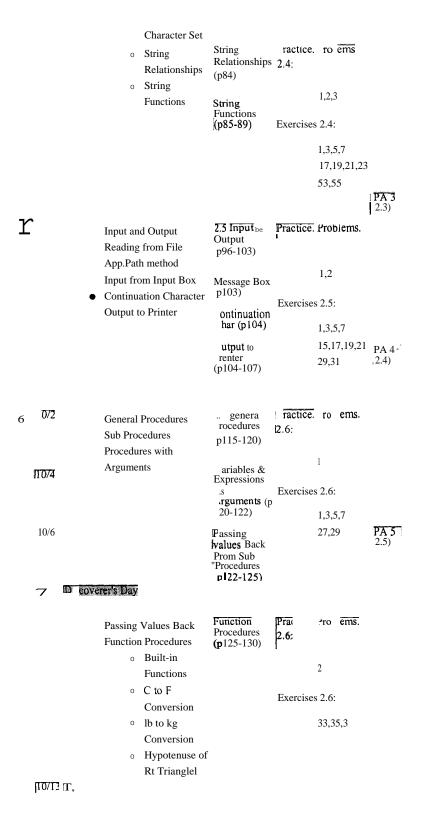
For "quick" questions the email is the simplest way to contact me. Feel free to drop in at My office during office hours or to set up an appointments outside those hours. (I am around my office usually in the afternoon.) When you need help in debugging programs, bring a copy of the code that is causing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

Return to Top

Chaminade CIS 150 University Class Schedule

<u>Syllabus</u>	Wk Date	•		Readings	Exercises	Due
Schedule	1	. introduction				
Resources		D 11 1			***************************************	
		Prelimir	Diagnostic		*	
Office flour:	0/36	U	Test	1.4 Intro to	***	
Record Check	8/30	0	Syllabus	VB (p22-25)		
Questionnaire			Terminology	15.		
150 Home			Local System	Programming Tools		
			Solving	(p25-26)		<u></u>
l lome		О	VB Problem			
			Solving			
			Algorithms			
		0	Pseudocode			
	1	I. VB Programm	ing Basics	Basic Objects	Practice Problems , 1:	
		Ohiects	& Events	(p36-47)		
			Objects	2.2 VB		
		0	Events	Events	Ir	
			Procedures	3	Exercises 2.1:	
	1978	Number	Type	2.3 Numbers (p63-69)	• 1,3,5	<u>PA</u> (2.1
		0	Arithmetic	(200-09)	• 30, 31, 37	(2.1
			Operations			
		0	Variables		Exercises 2.2:	
		0	Relational		• 1,3,5,7	
			Operators		• 37, 39, 41	
		0	Built-in		2 27,33, 41	
			Functions			
	3 19711 µ	I. VB Programm	ing Basics	, .	Practice Problems 2.3:	
		Object &	& Events			
		0	Objects			
		0	Event		Exercises 2.3:	
			Procedures			
		Number			• 1,3,5	
		0	Arithmetic Ops Variables		23,25,2735,39	
			Variables Built-in		₹ 55,55	
		0	Functions		,	
		String T				
			Literals &			
			Variables			
		0	Concatenation			
		0	Declaring			
			Strings			PA
		0	Scope of			PA (2.3)
		_	Variables Taxt Pox for			
		0	Text Box for			
			I/O	1	i	1

1 of 4 (09/12/2000 9·Δ0 ΔM



```
10/16 11. Controlling Program Flow
                                           3.1 Decision Practice. Problems.
                                            Structures
                                           (p 146-148)
                   Decision Structure
                         a If-Else
                                           Logical
                                           Operators (p152-155)
    10718
                            Structure
                                                          Exercises 3.1:
                         o Logical
                            Operators
                                                                    1,3,5
                         o Compound
                                                                   11,13,15
    10/20
                                                                   21,23
                            Conditions
                         o If-Else-If
                                                                   29,31,33
                            Structure
                                                   Loop Exercises 3.2
                   Do While Loop
                                           (p 164-166)
                   Algorithms with Do
                   While Loops
                                                                   1,3,5
   10/25
                                                                   7,9
                                                                   17,19,21
   10/27
                                           Input -& -
Output
(p96-100)
   10/30 eturning to 1 e 1/0
   11/3
   17/6
                                                          Practice. roblems.
         For...Next Loop
                                                          13.3:
                                              177-183)
                                                                   1,2
                                                          Exercises 3.3:
                                                                    1,3,5,7
                                                                   15
                                                                   17,19,21
                                              Creating Accessing
          IV. Arrays
                                                          Practice Problems
                                             rrays
                                            p200-209)
                   Declaring Arrays
                   Form-level Declaration
                   Form-Load Event
                                                          Exercises 4.1:
                                                                    1,3,5
                                                                   7,9
                                                                    19,21,23,25!
   1/10 eteran s ay t o Classes)
                                            4.2 Sorting & Practice Problems
12
                   Parallel Arrays
                                             earching
                                                           .2:
                   Array Algorithms
                                            p217-221)
                   Sorting
                                                                1,2
   11/15
                                                          Exercises 4.2:
                                                                    1,3,5,7
                                                                    17,19,21
   11717
                                                                                Loop),!,
```