Robert Maruyama

FD99
Prog

CIS 150: Introduct to Programming

Course Description

Through a combination of lectures, lab activities, and project assignments, this course will cover the basic concepts and techniques for 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 prerequisites for the course, a familiarity with the use of the microcomputer under the Windows 95 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
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

[Back 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
- 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 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.

Back 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. No credits will be given for a program that is past its due date by 4 days or more. If you have difficulty with your program, get help immediately so that you can stay on top of your assignments. The procedure for submitting assignments will be announced in the class.

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. 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.

[Back 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, readings, and 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.

[Back 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 F 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. University regulations limit the number of cuts a Freshman may take. Attendance will be taken at all class sessions. 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.

[**Back** 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
Class Participation:		50
Final Exam:		120
Total:		660

* 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 Syllabus

[Back to Top]

Criteria for Project Assin ments

Generally speaking, for each proramming assignment submit:

- **hardcopy of the code and**
- **image of the program interface**

An exception is PA No. l, which requires only the interface. How to produce the hard copy and the

image of the interface will be discussed in the class. (You an 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.

Yor projects will be evaluated on the following points.

- Correctness--does it work as advertized?
- 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 Control Option 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.

[Back 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.

[Back to Top]

Class Schedule
Fall Semester, 1999-2000

wk	Date	Notes	Text Book	Exercises	Due
1	8/30	I. Introduction 1. Preliminaries * Course Objectives * Terminologies * Computer System * Local System	* 1.1, 1.2 * 1.4		
	9/1	2. Problem Solving * AlgorithPs ms & udoc de * VB Prog amming II. Fundamentals of Programming in Visual Basic	* 1.5 **(p25,28)**	*Demo* (weeklyPay)	
	9/3	3. Objects & Events * VB Objects -- Text Box; Command Button; Label; Picture Box		*Class Exc* 2.1:Ex 30,31 2.1:Ex 36,37	
2	**9/6**	*Labor Day (no classes)*			
	9/8	* Event Procedures * Numbers * Functions -- Sqrt(); Into; ; Round0)	2.2 2.3 2.3 (p 70-71)	***Demo*** 3.2 Event Procedure 3.3 Balance	
	9/10			Class Exc **2.2:Ex** 37 **2.3:Exc** 57 **2.3:Exc** 59 **2.3:Exc** 63 **2.3:Exc** 64	Pa 1 (2.1)
3	9/13	4. Strings * Literals and Variables * Declaring Strings * Scope of Variables * ASCII Character Set * Option Explicit	**2.4 (p 77-83)** **2.4 (p 83-85)** **2.4 (p 88)**	**Class Exc.** **2.4:Exc 51** **2.4:Exc 52** **2.4:Exc 55**	
	9/15				
	9/17				***PA 2*** ***(2.3)***
4	**9/20**	* String Functions 5. Input and Output * Reading from File * App.Path method * Input from Input Box * Line-continuation character	**2.4 (p 85-89)** **2.5**	**Review:** **VB** Basics	
	9/22				
	9/24	***Test No. 1***			**PA** ***3*** **(2.4)**
5	**9/27**	6. General Procedures * Sub Procedures * Procedures with Parameters	2.6 **(116-122)** **2.6** **(122-125)**		
	9/29				
	10/1				_PA 4 (2.5)

	10/4	7. Function Procedures III. Controlling Program Flow	2.6 (p 125-130)		
	10/6	8. Decision Structures * If-Else Structure	3.1 (p 146-148)		
	10/8	* Compound Condition * Case Structure * If-Elseif Structure	3.1 (p 149-150)		EC 1
7	10/11	*Discoverer's Day (no classes)*			
	10/13'	9. Repetition Structures * Do Loop	3.2	Review: *Procedure* Decision Yruct.	
	10/15	*Test Na* **2**			PA 5 **(2.6)**
	10/1R	* Algorithms with Do Loops	3.3		
	10/20	* For-Next Loop			
	10/22				***PA 6*** ***(3.1)***
9	10/25	IV Arrays 10. Arrays * Declaring Arrays	4.1 (201-208)		
	10/27	* Form-level Declarations			
	10/29	* Form_ Load Procedure * Parallel Arrays	4.1 (p 209)		***PA 7*** ***(3.3)***
10	I1/1	*Veteran's Day (no classes)*			
	11/3	11. Sorting	**4.2 (p 217-222)**	**Review:** Loop Structures	
	11/5	***Test*** *No. 3*			
11	**11/8**	12. Binary Search * Median	**4.2(p 222-223)**		
	11/10				
	11/12				PA 8
12	**11/15**	13. Sequential Files * File Mode * Form Load	**4.3 (p 231-235)**		
	11/17				
	11/19				EC 2
13	**11/22**	* Sequential File * Modular Programming	4.3(p 231-235)	**Review:** Arrays	
	11/24		4.3 (p 235-237-ex. 3).		PA 9
	11/26	*Thanksgiving Holidays (11125-11126, no classes)*			
14	*11/29*	*Test No.* ***4***			
	12/1	14. Topics to be announced			
	12/3				
15	**12/6**	15. Review & Catching Up		**Review:** Files Finals	
	12/8				
	12/10				***PA 10*** ***EC 3***
16	*12/16*	***Final Exam: 12:45-2:45***			