Maruyama

R. Maruyama ? F0 '00

~~NO NAME~~

fM

Chaminade University of Honolulu

CIS 150: Introduction - 16-Programming

Syllabus

● Requirements ● Grading Tests Project ● Honesty ● Attendance ● Criteria

● Syllabus
Schedule
● 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

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

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 assigmnents. 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 x (4) = 240
Project Assignments:	25 x (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 University OF HONOLULU

CIS 150
Class Schedule

- Syllabus
- Schedule
- Resources
- Office Hour:
- Record Check
- Questionnaire
- 150 Home
- Home

Wk	Date	Topics	Readings	Exercises	Due
1	8/30	I. introduction Preliminaries o Diagnostic Test o Syllabus o Terminology o Local System Problem Solving o VB Problem Solving o Algorithms o Pseudocode	1.4 Intro to VB (p22-25) 15. Programming Tools (p25-26)		
	9/8	II. VB Programming Basics Objects & Events o Objects o Events Procedures Number Type o Arithmetic Operations o Variables o Relational Operators o Built-in Functions	.1 Visual Basic Objects (p36-47) 2.2 VB Events (p50-58) 2.3 Numbers (p63-69)	Practice Problems ', 1: Exercises 2.1: • 1,3,5 • 30, 31, 37 Exercises 2.2: • 1,3,5,7 • 37, 39, 41	PA 1 (2.1)
3	9/11	II. VB Programming Basics Object & Events o Objects o Event Procedures Number Type o Arithmetic Ops o Variables o Built-in Functions String Type o Literals & Variables o Concatenation o Declaring Strings o Scope of Variables o Text Box for I/O o ANSI (ASCII)	2.4 Strings (p 77-84)	Practice Problems 2.3: Exercises 2.3: • 1,3,5 • 23,25,27 • 35,39	PA 2 (2.3)

		Character Set ○ String Relationships ○ String Functions	String Relationships (p84) String Functions (p85-89)	ractice. ro ems 2.4: 1,2,3 Exercises 2.4: 1,3,5,7 17,19,21,23 53,55	PA 3 2.3)
r		Input and Output Reading from File App.Path method Input from Input Box ● Continuation Character Output to Printer	2.5 Input be Output p96-103) Message Box p103) ontinuation har (p104) utput to renter (p104-107)	Practice. Problems. 1,2 Exercises 2.5: 1,3,5,7 15,17,19,21 29,31	PA 4-` ,2.4)
6	0/2	General Procedures Sub Procedures Procedures with Arguments	.. genera rocedures p115-120) ariables & Expressions .s .rguments (p 20-122)	! ractice. ro ems. I2.6: 1 Exercises 2.6: 1,3,5,7	
	I10/4				
	10/6		Passing values Back Prom Sub "Procedures p122-125)	27,29	PA 5 2.5)
7	ID coverer's Day				
		Passing Values Back Function Procedures ○ Built-in Functions ○ C to F Conversion ○ lb to kg Conversion ○ Hypotenuse of Rt Trianglel	Function Procedures (p125-130)	Pra ro ems. 2.6: 2 Exercises 2.6: 33,35,3	
	I10/13 IT,				

8	10/16	11. Controlling Program Flow Decision Structure	3.1 Decision Structures (p 146-148)	Practice. Problems. 3.1:
	10718	a If-Else Structure o Logical Operators o Compound Conditions	Logical Operators (p152-155)	Exercises 3.1: 1,3,5 11,13,15
	10/20	o If-Else-If Structure Do While Loop Algorithms with Do While Loops	3. Loop (p 164-166)	21,23 29,31,33 Exercises 3.2 1,3,5
	10/25			7,9 17,19,21
	10/27			
	10/30	eturning to 1 e I/O	Input -& - Output (p96-100)	
	11/3	est		
	11/6	For...Next Loop	I-u zt cop 177-183)	Practice. roblems. 13.3: ● 1,2 Exercises 3.3: 1,3,5,7 ● 15 17,19,21
	!778	IV. Arrays Declaring Arrays Form-level Declaration Form-Load Event	1 Creating Accessing rrays p200-209)	Practice Problems 4.1: Exercises 4.1: 1,3,5 7,9 19,21,23,25!
	11/10	eteran s ay t o Classes!		
12		Parallel Arrays Array Algorithms Sorting	4.2 Sorting & earching p217-221)	Practice Problems .2: ● 1,2
	11/15			Exercises 4.2: 1,3,5,7 17,19,21
	11/17			

Loop),!,

13	11/20	Searching	Searching p222-225)		
	/24	Thanksgiving Holidays (No Classe			
14	1 1727 I1/29	Arrays & Sequential Files Sentinel Value	.3 Arrays & Sequential iles (p231-237)	Practice Problems k.3: ● 1 Exercises 4.3: 1,3 5 7,9,11,13 17	
15		● Problems with Sort and Search Algorithms ● Review & Catching Up			
	Fi	ii Exam			PAT