FD '02

Chaminade University of Honolulu

CIS 160: Introduction to Object-Oriented Programming

Syllabus
R. Maruyama

Course Description

Introduction to Object-Oriented programming using the C++ programming language, with examples drawn from business and other applications. Students should be acquainted with basic programming concepts, such as data types, variables, decision structures, and loop structures. Various aspects of the C++ syntax will be introduced throughout the course. Prerequisite: CIS 150 or equivalent

Course Objectives

At the end of the course students should be able to:

- Implement functions with greater ease
- Write programs of moderate size in C++ using algorithm chosen from alternatives
- Use top-down, step-wise refinement technique for structured programming
- Show understanding of abstraction in programming
- Have an understanding of the concepts of classes, objects, attributes and methods
- Write programs in terms of C++ predefined classes
- Use one-dimensional arrays in C++
- Differentiate strengths and weaknesses of different search and sort algorithms

Text Book

Structured & Object-Oriented Problem Solving Using C++ (3rd Edition), by Andrew C., Jr Staugaard. Prentice Hall, 2001. ISBN: 0130284513

Course Requirements

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

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

Tests & Quizzes

There will be two tests and one final exam. Refer to the Class Calendar for their dates. Test questions will consist of short questions similar to the ones in on-line quizzes (see the next paragraph) and C++ coding. When you foresee that you will not be able to make these dates for legitimate reasons, make *prior arrangements* with the instructor. In addition there will be short on-line quizzes at regular intervals, which are to be submitted via email. There will be no make up tests or quizzes for unexcused absences.

On-Line Quiz

The author of your text book provides sets of short on-line quizzes for each chapter. At regular intervals, you are required to take these quizzes, as specified in the Schedule page, and submit the results to the instructor. Here is the procedure.

1. Using the browser, go to http://cwx.prenhall.com/bookbind/pubbooks/staugaard2/
2. Scroll to the bottom of the page and *Select a Chapter*, as indicated in the class Schedule page--e.g., Ch 1, Ch 2, etc. Click *Begin.*
3. In the left colum, click either Multiple Choice (MP), Fill in the Blank (FIB), or True False (TF), as indicated in the class Schedule page. Complete the quiz.
4. If the results are satisfactory to you, submit the them to the following email address (Instructor): maruyamark@yahoo.com
5. You may try the quiz as many times as you wish before you send the results. However, the first page that is received by the instructor becomes your score for that quiz.

Project Assignments

Nine programming projects are scheduled for the term. Since learning the basics of programming in C++ is a major objective of the course, these projects are considered an indispensable part of the class. Each project is assigned points (maximum = 30 pts) according to its completeness, style, and timeliness. Refer to separate paragraphs for the project grading criteria. Programming assignments are available several weeks in advance of the due dates (at midnight via e-mail). All programming assignments which are late by 1 to 7 calendar days will incur a penalty of 30% (i.e., 21 pts max, instead of 30). Assignments which are later than 7 days will receive a maximum of 50%. If you foresee a valid reason that could cause you to hand in your assignments late--e.g., TDY, special assignment, etc.--you must get permission for special arrangements before the programming assignment is due.

Be conscientious in completing your assignments, since they are indispensable to learning programming fundamentals. Do not hesitate to seek help when you get stuck. When you are seeking help in debugging your programs, always accompany your questions with a hardcopy of your program listing or a copy of your algorithm written in pseudocode.

Programming assignments (PAs) should be submitted via e-mail in the following manner:

Send source code (with appropriate identification) and an image of an output screen of an assignment as e-mail attachment:

> To: maruyamark@yahoo.com
> Subject : CIS160PAx

Subject entry is important, if you want to be sure that your PA is received on time. In the Subject entry, x is the PA number. How to capture an image of an output screen will be explained in the class.

Grading Guidelines

The determination of the final course grades will be guided by the following distribution of course elements.

Project Assignments	30 x 9	270 pts
Tests	60 x 2	120 pts
Quizzes	6 x 10	60 pts
Finals Exam	100 x 1	100 pts
		-----------
		550 pts

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

A: = 90% B: = 80% C: = 70% D: = 60% F: < 60%

* A minimum of 6 completed projects is a necessary condition for a passing grade.

Project Grading Criteria

Your programming projects will be evaluated according to the following criteria:

1. Accuracy -- does the program perform as it was intended? Completeness -- does the program satisfy all the requirements stated in the original problem?
2. 2.Programming Style -- does it follow the style guidelines recommended for this class? Refer to the Programming Style guidelines for a list of common points to remember.
3. On Schedule -- was the project submitted on time?

Although you are encouraged to seek help from your TA and the instructor or discuss solutions with your classmates, the code you submit must be the result of your own work. Chaminade University General Catalog defines plagiarism as "the offering of work of another as one's own." (p. 51) Copying the work of others is considered a serious breach of contract in this class. Please read the section titled *Academic Honesty* regarding the penalties for plagiarism.

Academic Honesty

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 learns the basic programming skills by individual practice. You must distinguish between consulting your friends and discussing problems with them from copying other people's work.

Although you are encouraged to seek help from your TA and the instructor or discuss solutions with your classmates, the code you submit must be the result of your own work. *Chaminade University General Catalog* defines plagiarism as "the offering of work of another as one's own." (p. 51) Copying the work of others is considered a serious breach of contract in this class.

The penalty for copying from someone else's program or parts there of is a grade of 0 for all parties involved for the first offense; and an F in the course for the second offense.

Programming Guidelines

In a class like this, where you are learning the fundamentals of programming, you must do your programming work by yourself. Although software development in the real world is a group activity, where each member of a development team must be able to communicate with others at all levels, you need to practice writing program every chance you have.

Your programs will be more readable and understandable, both to you and to others (and to your grader, in this case), if the code is organized and written in a uniform manner. Although standards for programming styles and conventions can sometimes seem subjective, a certain degree of discipline and uniformity is necessary in any software development activity. Your source code should follow a certain style, such as documenting comments, use of upper and lowercases for identifiers, indentation with control structures, blank space between symbols, and blank lines between statements.

The following is a highlight of some of the common points. Read the section in your textbook on Programming Style Guidelines (p 587ff) for a more complete description of style conventions we will be using in this class.

a. Tabs are set every 3 spaces.
b. Variables and function names begin in lowercase.
c. In general, each declaration occupies one line.
d. There are spaces after keywords and between binary operators (e.g., c = a + b; not c=a+b;).
e. Braces must line up.
f. Indent the body of a decision structure or a loop structure.
g. No "magic numbers" should appear in your program.
h. Vertical spacing--insert a blank before and after a loop structure or a decision structure. Separate major sections in the program--e.g.,

input section, process section, output section--with a blank line.

i. Comments should help the programmer. Avoid commenting too much or commenting too little. Major code sections in a program deserve explanations.
j. In general, global variables should be avoided.
k. Every program should begin with documenting comments. See handouts for examples.

Attendance

Regular class attendance is important, since you are responsible for all topics covered in the class. Please make sure that you have someone from whom you can obtain notes, in case you miss a class. Generally speaking, there will be no make-up tests. When there are legitimate reasons for not meeting scheduled test or assignment dates, arrangement must be made before those scheduled dates.

Chaminade University of Honolulu

CIS 160: Introduction to Programming

Schedule

● Syllabus ● **Schedule** ● Resources ● Records ● Office ● Home

Wk	Date	Topics	Readings	PowerPt	PA	On-line Quiz
1	8/26	• Introduction • Hardware System • Software System • Operating System o Machine language o Language Translator o High-level language • *Demo (IDE)*	Ch 1 p.2 - 24	Ch01.pdf		
	8/28					
	8/30					
2	9/2	Labor Day (No Classes)				
	9/4	• Programming Steps o Define problem. o Plan problem solution o Code program o Test and debug o Document program • Structured Programming o Abstraction (top-down) o Step-wise Refinement • OOP	Ch. 2 p. 31 - 48	Ch02.pdf		No 1: Ch 1 (MC)
	9/6				• PA 1	
3	9/9	• Data Types • Data Hierarchy in C++ • Primitive Data Types • Constants and Variables	Ch. 3 p. 52-73	Ch03.pdf		No 2: Ch 1 (FIB)
	9/11					
	9/13				• PA 2	
4	9/16	• Strings • String Operations	p. 84 - 90	Ch04.pdf		No. 3: Ch 2

		• ***Demo (Total)***				(MC)
	9/18					
	9/20					
5	9/23	Test No. 1 (Review Questions)				
	9/25	• I/O Objects • Formatting methods o Field Width o Decimal Points o Left-justified • Using cin Object o Reading char and String • ***Demo (Interest)***	Ch. 4 p. 98-138			No. 4: Ch 3 (MC)
	927				• PA 3	
6	9/30	• Arithmetic Operators • Order of Precedence • Increment Operator • Conversion Functions • Math Functions	Ch 5 p. 170-181	Ch05.pdf		No. 5: Ch 3 (FIB)
	102					
	10/4				• PA 4 • Extra Credit	
7	10/7	• Relational Operators • Logical Operators • Dicision Structures o If Statement o If-Then-Else Statement o Multiple-If statement o Switch Statement • ***Demo (Pay Increase)*** • ***Demo (Convert Grade)*** • ***Demo (Convert Grade 2)*** • ***Demo (Memu Driven)***	Ch 6 p. 210-249	Ch06.pdf		No. 5: CH 4 (MC)
	10/9					
	10/11					
8	10/14	Discoverer's Day (No Classes)				
	10/16	• While Loop • Do While Loop • ***Demo (While Loop)***	Ch 7 p. 264-282	Ch07.pdf		No. 6: Ch 4 (FIB)

	10/18				• PA 5	
9	10/21	• For Loop • ***Demo (For Loop)***	p. 282-291			No. 7: Ch 5 (MC)
	10/23					
	10/25				• PA 6	
10	10/28	• Nested Loop • ***Demo (Pattern)***				No. 8: Ch 5 (FIB)
	10/30					
	11/1					
11	11/4	• Benefits of Functions • Void Functions • Non-void Functions • ***Demo (Functions)*** • ***Demo(num2letter)***	Ch 8 p. 316-328	Ch08.pdf		No. 9: Ch 6 (MC)
	11/6					
	11/8				• PA 7	
12	11/11	• Value vs Reference Paramenters • Function Protocol • Recursion • ***Demo (Recursion)***	p. 328-341 p. 363-369			No. 10: Ch 6 (FIB)
	11/13					
	11/15					
13	11/18	• Array Definition • Assignming Values to Array • Visiting Each Element with For Loop • Passing Arrays to Functions • Insertion Sort • ***Demo (Array)***	Ch 9 p. 386-498 p. 408-412 p. 419-424	Ch09.pdf		No. 11: Ch 7 (MC)
	11/20	Test No. 2				
	11/22	• Binary Search	p. 424-		• PA	

			430		8	
14	11/25	Ch 10 • Class Definition • Member Functions • Data Members • ***Demo (Class)***	Ch 10 p. 440-469	Ch10.pdf		No. 12: Ch 7 (FIB)
	11/27					
	11/28-/29	Thanksgiving Recess (No Classes)				
15	12/2	• Program Using a Class (PA 9) • Review				No. 13: Ch 8 (MC)
	12/4					
	12/6				• PA 9	
16	12/12	**Final Exam:** (10:30-12:30)				