
Lamar University

COSC 1373-01, Summer 2003
Monday~Friday, 12:45~2:05 PM, Maes 109

<http://hal.lamar.edu/~licc/COSC1373/index.html>

Principles of Computer Science I (C++)

Instructor: Dr. Chung-Chih Li
<http://hal.lamar.edu/~licc>
licc@hal.lamar.edu

Office Hours: Maes 69
(409) 880-8748
MWF 10:30~11:30 am

Course Description and Purpose:

This course is designed to give you a concrete concept of computing by modern computers in terms of programming language. After a brief introduction to the major components and underlying principles of computers, we will study one of the most important programming language, C++, and use it to solve some basic computing problems. The topics to be covered are the most important and basic features of C++ including elementary data types, input/output, functions and their parameters, program control, arrays, records and files, modular design, and structured programming. Since C++ is an *object oriented* programming language, we will introduce some basic concepts of OOP design such as *classes*, *public and private functions*, *friend classes*, and, if time allows, *inheritance*). However, we will not particularly focus on such programming paradigm due to the time constraint. Also, we will learn how to analyze the complexity of a program in terms of its efficient of running. I like to introduce some basic features of Visual C++ graphics, but it is not very likely to have time for this.

Programming Environment:

We will use **Microsoft Visual C++ 6.0** as our standard programming environment. The PC's in Maes 215 Lab have the required softwares installed. You shall be familiar with the environment from their lab works; including editing, compiling, linking, and debugging C++ programs. We will not cover them in the regular class. The introductory version that comes with the textbook is sufficient for the assignments. You can walk in the lab any time at your convenience to work on their assignments. Coming to the lab is not mandatory as long as you can complete the assignments by using their own computers on time.

Prerequisites:

There are no specific prerequisites needed for this course. Nevertheless, some basic concepts of algebra, sets, and functions are needed, but we can fix and clarify any misconception whenever needed in the class. There is no need to have any experience of programming languages, but a certain experience of working with computers may help you to take off at the beginning without too much frustrating.

Textbook:

- *Absolute C++*, by Walter J. Savitch, Addison Wesley Publishers, 2002

We will cover most of the materials introduced in Chapters 1 to 9 of the textbook. If time allows, we will briefly introduce recursion and inheritance from Chapters 13 and 14.

Reference Books:

- *Computer Science Tapestry – Exploring Programming and Computer Science with C++*, by Owen L. Astrachan, McCraw Hill Publishers
- *Practical C++ Programming*, by Steve Oualline, O'Reilly Publishers

Grading Policy: (600 is considered the perfect score)

Points	Grade	
540 ~	A	Excellent
420 ~ 539	B	Good
300 ~ 419	C	Satisfactory
200 ~ 299	D	Passing
0 ~ 199	F	Failure

Examinations and Dates: (300 points)

All tests are accumulative, closed book, and indispensable. No makeup test will be given unless a documented absence is authorized by the university.

Midterm 1	100 points	10th class	Jun. 13, Friday, 12:45~2:00 pm
Midterm 2	100 points	20th class	Jun. 27, Friday, 12:45~2:00 pm
Final Exam	100 points	26th class	Jul. 8, Tuesday, 12:45~2:00 pm

- Every student is allowed to bring a self-prepared crib in one letter-sized paper to the test.

Assignments: (250+10 points)

Eight programming assignments are expected. Each assignment of the first six will carry 30 points towards students' final scores; 40 points for each of the last two. Late works will be graded with penalty.

Students are encouraged to discuss assignments and help each other. However, this does not include **either entirely or partially copying or modifying** someone else's programs.

Attendance: (50+? points)

Each attendance, if taken, contributes 5 points towards your final scores. In other words, an absence costs 5 points.

Academic Honesty:

Cheating, plagiarism, collusion, abuse of resource materials, and their consequences are defined and described under the chapter of Academic Affairs in the *Student Handbook*.

Giving away academic works for assignment offered for credit to other students will be considered as academic dishonesty, and will receive the same penalty.

Special Notes to The Class:

1. Backup your works. "My dog ate my disk!!" is not a good excuse.
2. We will have a seat arrangement for this class. This helps me to remember your names and check roll quickly. You should have a seat where you feel most comfortable; but where you sit in the class will not affect your grade in any way.
3. To Computer Science Major students:

This is the first course of the three-course-series, COSC 1373-1374-2371, which is required for CS major students. In the second course COSC 1374, we will assume that students already have certain maturity of programming skill in C++ and cover more advanced features, especially OOP design, of C++. The first two courses are designed to build up students' muscle for taking the third course, COSC 2371, Data Structure and Algorithm Analysis. By the time students finish the three courses, their skills in programming should be well enough to handle most entry-level programming jobs. Also, with a solid knowledge of C++, students will find it easy to learn other programming languages by themselves for their future studies and jobs. For example, C++ programmers can easily pick up Java by themselves due to the great similarity between the two languages.

Tentative Topics and Schedules for COSC 1373-01

Week	Topics	Reading
1: Jun. 2 ~ Jun. 6	<p>Introduction to computer, computer science, programming language, program design, binary number system.</p> <p>Introduction to C++, literals, variables, assignments, I/O, program style, Visual C++ 6.0 basic.</p> <p>Expressions, boolean expressions, conditionals, flow of control, if and nested if statements, loops.</p>	<p>lecture note</p> <p>1.1~1.5</p> <p>2.1~2.3</p>
2: Jun. 9 ~ Jun. 13	<p>Loops and nested loops, functions and parameters, scopes.</p> <p>Nested scopes, parameters, call-by-value, call-by-reference.</p> <p>Overloading and debugging techniques. (Midterm 1)</p>	<p>2.3~3.3</p> <p>3.3~4.1</p> <p>4.2~4.3</p>
3: Jun. 15 ~ Jun. 20	<p>Homogeneous aggregates: arrays, arrays in functions, searching and sorting an array.</p> <p>Multidimensional arrays, Heterogeneous aggregates: struct, structs in functions</p> <p>Terminologies of OOP, classes, comparison between structs and classes, member functions, private and public variables and methods in classes.</p>	<p>5.1~5.3</p> <p>5.4~6.1</p> <p>6.1~6.2</p>
4: Jun. 23 ~ Jun. 27	<p>Constructors, classes in classes, case study: BankAccount Class and Vectors.</p> <p>Operator overloading, class's friends and friend member functions.</p> <p>String operations, string in terms of arrays, string in terms of classes. (Midterm 2)</p>	<p>7.1~7.2</p> <p>8.1~8.3</p> <p>9.1~9.3</p>
5: Jun. 30 ~ Jul. 3	<p>Binary search and Recursion.</p> <p>More on Recursion, computer graphic in C++.</p> <p>Catch-up, review for the final. (Thanksgiving Holiday).</p>	<p>13.1~13.3</p> <p>lecture note</p>
6: July 7, 8	<p>Prepare for the final.</p> <p>Final Examination.</p>	