

---

# Lamar University

## COSC 1373, Fall 2002

### Principles of Computer Science I (C++)

---

Instructor: Dr. Chung-Chih Li  
Office: 69 Maes, tel: (409) 880-8748  
URL: <http://hal.lamar.edu/~licc>  
E-mail: [licc@hal.lamar.edu](mailto:licc@hal.lamar.edu)  
Office Hours: MWF 11:10 AM ~ 12:10 PM or by appointment

**Class meeting times and place: (Attendance will be taken impulsively)**

Section 01	TuTh	9 : 30 ~ 10 : 45	AM	} both meet at Maes 111
Section 03	MWF	8 : 00 ~ 8 : 50	AM	

**HomePages of the course:**

**TBA**

These webpages contain detailed information about this course, assignments, due dates, and most recent announcements.

**Course Description and Purpose:**

This course is designed to give students a concrete concept of computing by modern computers. After a brief introduction to the major components and underlying principles of computers, we will emphasize on the problems that by their natures can be solved by computers. In other words, the purpose of this course is to learn how to use programming languages to program the computer so it can find the solutions to the problems. In particular, we choose C++ as the programming language for our study. The topics to be covered are the most important yet 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 the concepts of *classes* (and, if time allows, *inheritance*) in C++, but 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.

**Programming Environment:**

We will use **Microsoft Visual C++ 6.0** as our standard programming environment for our C++ programs. The PC's in Maes 215 Lab have the required softwares installed. Students 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. Students can walk in the lab any time at their convenience to work on their assignments. The introductory version of Microsoft Visual C++ 6.0 that comes with the textbook is sufficient for most of the assignments. Coming to the lab is not mandatory as long as students 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. Students taking this course do not have to have any experience of programming languages. However, a certain experience of working with computers may help to take off at the beginning without too much frustrating.

**Special note to Computer Science Major students:**

This is the first course of a three-course-series, COSC 1373-1374-2371, which is required for CS major students. The second course COSC 1374 will assume students having certain maturity of programming skill in C++ and will shift the programming environment to Unix to study more advanced futures of the language. The first two courses are designed to build students' muscle for taking COSC 2371 Data Structure and Algorithm Analysis. By the time students finish the three courses, they should be able to program in C++ well enough for most entry-level jobs. Also, with a solid knowledge of C++, students will find it easy to learn other programming languages by themselves for their future studies. An obvious example is that a C++ programmer can easily transfer to Java due to the great similarity between the two languages.

**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, or, alternatively, some graphic features of Visual C++.

**Reference Books:**

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

**Examinations and Dates:** (400 points)

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

			Section 01 (TTh)	Section 03 (MWF)
Midterm 1	100 points	6th week	Sep. 24, 10:00 ~ 10:45 AM	Sep. 23, 8:05 ~ 8:50 AM
Midterm 2	100 points	12th week	Mov. 5, 10:00 ~ 10:45 AM	Mov. 4, 8:05 ~ 8:50 AM
Final Exam	200 points	17th week	Dec. 10, 8:00 ~ 10:30 AM	Dec. 9, 8:00 ~ 10:30 AM

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

**Assignments:** (150 points)

Five programming assignments are expected. Each assignment carries 30 points towards students' final scores (the perfect score is 600 points). Late works will be graded with penalty; -3 points per day after the due date.

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 or more)

Each attendance, if taken, contributes 5 points towards students' final scores. In other words, an absence on the day the roll is checked costs 5 points.

**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

**Academic Honesty:**

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

Students giving away academic works for assignment offered for credit to other students working on the same assignment will be considered as guilty as academic dishonesty, and will receive the same penalty.

**Special Notes to The Class:**

- Backup your works. "My dog ate my disk!!" is not a good excuse.
- Since the more often I take the attendance the more points you may get, you will be better off if I can check the roll efficiently on a daily basis. In order to take the attendance in an efficient way, we will have a seat arrangement for this class. This also helps me to remember your name. Your seat will be fixed to where you sit in the second class of the semester. So, before the second class begins, please decide a seat (negotiate with your classmates, if necessary) where you feel comfortable to sit in for the rest of the semester. However, where you sit in the class will not affect your grade in any way.
- Although there is no team-work project in this course, every student is encouraged to joint a study group for exchanging ideas and discussing materials covered in this course. A typical study group should have 3 to 5 students who can meet conveniently and regularly during the semester outside the classroom. If you can't form a study group by the end of the second week, the instructor will do that for you. None of the activities of the study groups will directly connect to the grade.
- Every student has to prepare a letter-sized Manila folder with the course number and your name written on the tab of the folder. After the study group is determined, write the name list of your group on the page of the folder (facing inside), and give the folder to the instructor. The folder will be used to hold your assignments, tests, attendance, and any personal information regarding the class.

## Tentative Topics and Schedules for COSC 1373

Week	Topics	Reading
1: Aug. 21	Introduction to computer, computer science, programming language, program design, binary number system.	lecture note
2: Aug. 26	More on binary number system, introduction to C++, literals, variables, assignments, I/O, program style.	1.1~1.5
3: Sep. 2	Expressions, boolean expressions, conditionals, flow of control, if and nested if statements, loops.	2.1~2.3
4: Sep. 9	Loops and nested loops, functions and parameters, scopes.	2.3~3.3
5: Sep. 16	Nested scopes, parameters, call-by-value, call-by-reference.	3.3~4.1
6: Sep. 23	<b>(Midterm 1)</b> , Overloading and debugging techniques.	4.2~4.3
7: Sep. 30	Homogeneous aggregates: arrays, arrays in functions, searching and sorting an array.	5.1~5.3
8: Oct. 7	Multidimensional arrays, Heterogeneous aggregates: struct, structs in functions	5.4~6.1
9: Oct. 14	Terminologies of OOP, classes, comparison between structs and classes, member functions, private and public variables and methods in classes.	6.1~6.2
10: Oct. 21	Constructors, classes in classes, case study: BankAccount Class and Vectors.	7.1~7.2
11: Oct. 28	Operator overloading, class's friends and friend member functions.	8.1~8.3
12: Nov. 4	<b>(Midterm 2)</b> , string operations, string in terms of arrays, string in terms of classes.	9.1~9.3
13: Nov. 11	Binary search and Recursion.	13.1~13.3
14: Nov. 18	More on Recursion, computer graphic in C++.	lecture note
15: Nov. 25	Catch-up, review for the final. (Thanksgiving Holiday).	
16: Dec. 2	Prepare for the final.	
17: Dec. 9, 10	<b>Final Examination.</b>	