



Fundamentals of Computation

CS 102 (2 Units)

Overview

This course introduces students to the fundamental concepts of algorithmic thinking as a primer to programming. The course introduces the mathematics required of programming as well as the creative problem solving techniques required to analyze a problem and produce an algorithm. These techniques are put into practice over the course of the semester with an introduction to programming based in C++. A lecture/lab course format will be employed to provide hands-on experience and active learning techniques.

Learning Objectives

Upon completion of this course students will be able to:

1. Understand how to analyze a problem and develop a computational solution
2. Understand an algorithm as a description of how to solve a problem
3. Understand how to write, compile and run a computer program
4. Understand the basic mathematics required of programmers
5. Understand the way computers represent and operate on data
6. Understand basic programming concepts: variables, control structures, loops, and arrays
7. Understand how to interpret written program specifications and develop a programmatic solution.

Prerequisite

None.

Instructor Info

Professor Mark Redekopp
Office: EEB-222, Phone: (213) 740-6006
E-mail: redekopp@usc.edu
Office Hours: See course website

Administrative Information

Course Website

<http://bytes.usc.edu/cs102>

All course info is posted here. **All policies are officially posted here with this document serve as a synopsis. You should refer to the website from now on.**

Blackboard

<http://blackboard.usc.edu> (use your e-mail username and password)

Online Q&A

<https://piazza.com/class/jazwuy2y9ry345>

TA & Mentor Info:

This course has a large support staff consisting of TAs and many undergraduate "course producers". They are meant to guide you and help you and will have a wide array of availability in addition to the instructor. Please utilize their help. **See course website for their availability.**

Textbook C++ For Everyone, 2nd Ed., Cay Horstmann, J Wiley and Sons, 2012 (ISBN: 978-0470927137)

Required Materials A computer capable of running the class virtual machine image (see course website)

Grading The following point structure will be used in determining the grade for the course. Final grade will be based upon the total points received, the highest total in the class, and the average of the class.

Lab Participation	5%
Homework	30%
Midterm Exam	30%
Final Exam	35%

Homeworks Roughly each week a homework will be assigned and is due on the following Thursday evening. Homeworks should be done **individually** unless otherwise stated. Copying or sharing details of your solution with another student IS an academic integrity violation. Please consult course staff or Piazza for help. Assignments may be submitted 1 day late for max 50% credit. Beyond 24-hours, there is no late submission (regardless of laptop/wireless issues). Finish the assignment early, submit it a few hours ahead of time to allow for such problems.

Policies Statement for Students with Disabilities

Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me (or to TA) as early in the semester as possible. DSP is located in STU 301 and is open 8:30 a.m.–5:00 p.m., Monday through Friday. The phone number for DSP is (213) 740-0776.

Statement on Academic Integrity

Copying (and then modification) of any portion of code from Internet sources or fellow students is prohibited unless cleared with the instructor. In addition, working together on a programming assignment is also a gray area. If your code resembles that of your friend's too closely, that will be considered a VIOLATION.

Please review the course website for what IS and IS NOT acceptable collaboration.

USC seeks to maintain an optimal learning environment. General principles of academic honesty include the concept of respect for the intellectual property of others, the expectation that individual work will be submitted unless otherwise allowed by an instructor, and the obligations both to protect one's own academic work from misuse by others as well as to avoid using another's work as one's own. All

students are expected to understand and abide by these principles. *Scampus*, the Student Guidebook, contains the Student Conduct Code in Section 11.00, while the recommended sanctions are located in Appendix A: <http://www.usc.edu/dept/publications/SCAMPUS/gov/>. Students will be referred to the Office of Student Judicial Affairs and Community Standards for further review, should there be any suspicion of academic dishonesty. The Review process can be found at: <http://www.usc.edu/student-affairs/SJACS/>.

Emergency Preparedness/Course Continuity in a Crisis

In case of a declared emergency if travel to campus is not feasible, USC executive leadership will announce an electronic way for instructors to teach students in their residence halls or homes using a combination of Blackboard, teleconferencing, and other technologies.

Weekly Schedule

	Mon	Wed	Readings
Week 1	Computer System Overview	Data Representation and Storage	Ch. 1.1-1.4
Week 2	Holiday	Languages, Syntax and Compilation	Ch. 2.1-2.3
Week 3	Types, Constants, Variables, and Expressions	More Expressions, Assignment, statements & Sequence of execution	Ch. 2.1-2.3
Week 4	Functions Part 1	Algorithmic Thinking with Expressions & Functions	Ch. 5.1-5.4
Week 5	Conditional (If) Statements	Algorithmic Thinking with If statements	Ch. 3
Week 6	Iterative (while-loop) structures	Algorithmic Thinking with While loops	Ch. 4.1, 4.2, 4.4
Week 7	Holiday	More iterative (for-loop) structures	Ch. 4.3
Week 8	Algorithmic Thinking with for loops (sequences and series)	No Lab - Midterm (Install the VM on own)	Class Notes
Week 9	Return and Review Midterm; Debugging 1	Debugging 1 (cont.)	
Week 10	Nested Loops	Algorithmic Thinking with nested loops	Ch. 4.8
Week 11	Arrays and Collections	Common Array Tasks	Ch. 6.1-6.2
Week 12	Functions 2	Functional Decomposition	Ch 5
Week 13	Passing Arrays As Arguments	Strings	Ch 6.3, 2.5, 7.3
Week 14	Debugging 2	Highlighted Algorithm	Class Notes
Week 15	Linux Commands	Review	Class Notes
FINAL		(9 a.m.) Fri. May 4 th – 8 a.m. (10 a.m.) Mon. May 7 th – 8 a.m.	