USC
SCHOOL OF
ENGINEERING

Fundamentals of Computer Programming

CS 101 (3 Units)

Overview

This course introduces students to the field of computer science and engineering. An overview of the disciplines within computer science such as networks, AI, robotics, graphics, and computer architecture will be integrated throughout the course. Starting from first principles of computer organization, students will receive a foundation in programming focusing on C/C++. Fundamental programming concepts along with current issues such as parallelism and embedded systems will be covered through relevant programming projects. The course will culminate in a comprehensive programming assignment and/or a team-based robotics project that integrates the concepts taught in the course. 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. Define and discuss the disciplines of computer science with the intention of choosing future courses that are applicable to the student's personal learning goals.
2. Understand the key hardware components in a modern computer system and how software is mapped to the HW.
3. Use a computer to solve problems by developing simple algorithms and then implement them using a specific programming language
4. Implement key algorithms within the field
5. Understand and determine the computational complexity of simple algorithms
6. Write computer programs using conditional and iterative structures, functional decomposition, and basic parallelization techniques
7. Select an appropriate basic data structure (e.g. arrays) and access methods (e.g. pointers)
8. Understand basic object-oriented principles.
9. Design non-trivial embedded software to control a robot to navigate and interact within a controlled environment.

Prerequisite None.

Lecture M,W 12:00-1:50 p.m. in RTH 105

Discussion TBA

Textbook Info C++ Programming: Program Design Including Data Structures, **6th** Ed. D.S. Malik, Course Technology, 2011 (ISBN 978-1133526322)

Instructor Info Professor Mark Redekopp
Office: EEB-222, Phone: (213) 740-6006
E-mail: redekopp@usc.edu
Office Hours: M,W: 10-11:30, 2-3 and T: 11-12

TA & Grader Info: Ali Khodaei (khodaei@usc.edu)
Grader: TBA

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.

Participation & Activities	10%
Programming Assignments	30%
Midterm Exam	25%
Robotics Project + Report	10%
Final Exam	25%

Participation & Activities Unlike some traditional classroom settings where the instructor talks and students listen, we will attempt to create a classroom environment where the instructor facilitates active student participation in their own learning process. Students are expected to set their own learning goals (i.e. be curious) and then actively pursue those goals **both in and out of the classroom** through personal study, programming, and in-class activities. Simply showing up to class is not enough; come to class ready to think, ask questions, and work with your fellow students.

Small in-class and out-of-class activities (both individual and group-based) will be provided to help facilitate achievement of learning goals. Activities will be graded on a scale of 0-4 as follows:

- 4 = Excellent = Demonstrates deep understanding through use of relevant concepts, consideration/analysis of possible improvements, and awareness of the limitations of your solution as well as a high-degree of effort in the presentation and documentation of your solution
- 3 = Good = Demonstrates sufficient understanding through the use of relevant concepts to solve the problem. May indicate only moderate effort in the presentation and documentation.
- 2 = Novice = Important relevant concepts were ignored or used improperly. Moderate effort was made in the presentation and documentation.
- 1 = Unsatisfactory = Solution was poorly developed and ignored relevant concepts as well as any analysis of the

solution's capability, improvement, or limitation. Low-effort was made in the presentation and documentation of the solution

- 0 = Incomplete

Students may miss 2 activities due to sickness or absence without penalty.

Programming Assignments

Programming assignments are larger more comprehensive assignments that should challenge you to integrate several programming concepts. **They are to be completed individually unless otherwise noted**. (A few group assignments may be scattered throughout the semester). A separate document will be provided listing the expectation for the submission, style, and documentation of programming assignments as well as rubrics for how assignments will be graded. Copying (and then modification) of any portion of code from Internet sources or fellow students is prohibited unless cleared with the instructor. See the Statement on Academic Integrity.

Robotics Project and Report

Students will work in groups of two to complete a robotics project related to sensing and navigation. Project and report guidelines will be provided at the appropriate time.

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

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/.

Introduction to Programming for Computer Scientists

CSCI 101 (3 Units)

Course Outline

(All Programming Assignments [PA's] listed indicate the date assigned)

Week 1 – Overview of Computer Science and Computer Organization

- Introduction to the CS and CECS curriculum and field
- Basic Computer Organization, Program Execution, Data Storage/Representation

Activity: Programming Environment and Tools + First Program: Learn basics of programming environment, understand the concept of compiled program, text editors, debuggers, etc.
Reading: Chapter 1

Week 2 – Basic Program Design and Abstractions + Programming Environment

- Program structure + Simple I/O
- Algorithms, Big-O notation and complexity

Activity: Compiler and Run-Time Errors
Activity: Algorithms and time complexity exercise + Research algorithms in a chosen area of interest
Reading: Chapter 2 and 3

Week 3 – Algorithmic Thinking and C Control Structures

- Conditional and Iterative Statements

Activity: Develop play rules for a game-
PA1: Students will program a simulation of random process and understand how to use iterative and conditional structures in the process.
Reading: Chapter 4 and 5

Week 4 – Program Decomposition and Functions

- Functions

Activity: Write the program that implements a linear as well as a binary search of an integer array
PA2: Develop a program for a simple game. Decompose the problem into tasks/functions.
Reading: Chapter 6

Week 5 – 1-D Arrays, Pointers, Dynamic Memory Allocation, and C Strings

- Dynamic memory allocation

PA3: String Manipulation: Students will design a program to perform string matching of certain patterns that require students to understand array operations and strings.
Reading: Chapter 9 and 13a

Week 6 – 2-D Arrays and File I/O

- 2D Arrays and File I/O

Activity: Implement string search using a DFA approach and measure improvement
PA4: Chroma Key (Green-screen) Image Processing: Students will perform the green-screen operation on a set of images requiring them to access and manipulate 2D arrays and File I/O operations.
Reading: Chapter 9 and 13a

Week 7 – Parallel Programming Techniques

- Threads & OpenMP

Activity: Parallelize string search
PA5: Parallelizing Matrix Operations – Students will parallelize data independent matrix operations such as the green screening or other image manipulations.
Reading: Class Notes

**** MIDTERM ****

Week 8 – More Parallel Program and Introduction to Objects & ADT's

- Basic Synchronization (Locks, Condition Variables)
- Structs

Activity: Parallelize Monte-Carlo simulation
Reading: Chapter 10 and 11

Week 9 –Classes

- Classes

Activity: Linked List Implementation
PA6: Social network modeling
Reading: Chapter 13b and 17

Week 10 – Data Structures and STL

- Vectors & Deques

Reading: Chapter 14 and 22

Week 11 – Selected Algorithms & CS Disciplines

- Graphs and Traversal Algorithms

PA7: Implement an algorithm of choice on your social network
Reading: Chapter 21

Week 12 – Introduction to Embedded Systems and Robotics

- Programming the robot (Language differences)
- I/O
- Robotics Project Intro

Activity: Robot Motion – Students will program their robot to perform specific maneuvers

Week 13 – Robot Project 1

- Sensors

Activity: Robot Sensing – Students will program their robot to use sonars, IR sensors, and photosensors to navigate their robot

Week 14 – Robot Project 2

- Guest Lectures + Open Lab for Project

Project: Robot Obstacle Course and/or Collaborative Team Maneuvers

Week 15 – Robot Project 3
Project Due/Demo Day

**** Final – December 10th at 11 a.m.****