Las Positas College
3000 Campus Hill Drive
Livermore, CA 94551-7650
(925) 424-1000
(925) 443-0742 (Fax)

Course Outline for CS 7

INTRODUCTION TO COMPUTER PROGRAMMING CONCEPTS

Effective: Fall 2005

I. CATALOG DESCRIPTION:
CS 7 — INTRODUCTION TO COMPUTER PROGRAMMING CONCEPTS — 3.00 units

An introductory course in computer programming for nonscience majors and for students requiring additional preparation before taking Computer Science 1. Hardware, system software basics, the history of computing, computer ethics, basic computer operations, number systems, design of algorithms, pseudocoding, flowcharting, and programming constructs such as variables, expressions, input/output, decision-making, loops.

2.50 Units Lecture 0.50 Units Lab

Grading Methods:
Letter or P/NP

Discipline:

	MIN
Lecture Hours:	45.00
Lab Hours:	27.00
Total Hours:	72.00

II. NUMBER OF TIMES COURSE MAY BE TAKEN FOR CREDIT: 1

III. PREREQUISITE AND/OR ADVISORY SKILLS:

IV. MEASURABLE OBJECTIVES:

Upon completion of this course, the student should be able to:

1. identify the various parts of modern computational systems and their use;
2. identify the differences between system software and application software;
3. describe the ethics of computer use;
4. perform basic computer operations using features of Windows and the desktop;
5. use different number systems and convert numbers from one base to another;
6. diagram the history of computation and programming languages;
7. describe the concept of programming;
8. explain the notion of an algorithm;
9. draw a flowchart of a computer program;
10. list and develop steps that are involved in program development;
11. design correct and efficient algorithms to solve programming problems;
12. write pseudocode to solve programming problems;
13. draw flowcharts to indicate the flow of algorithms;
14. construct small programs using various elements, such as variables, I/O, if-else, loops, functions, expressions, and parameters.

V. CONTENT:

A. Computer Systems
   1. System overview
   2. Distinction between hardware, software, and firmware

B. Computer Ethics
   1. The 10 commandments of Computer Ethics
   2. The role of government and business

C. Hardware
   1. Overview of the 5 major components including input devices, output devices, central processing unit, primary memory, and secondary storages
   2. Input devices: How to efficiently use the keyboard and mouse
   3. Output devices: How to obtain hard copies using the printer
   4. CPU: control unit, ALU, registers, and buses
   5. Memory: RAM vs. ROM
   6. Storage: Use and care of diskettes, zip disk, jump drives and other storage devices
   7. Simple circuits

D. Software
   1. Distinction between system and applications software
   2. System software basics

3. Microsoft Windows: logging in; shutting down; use of start button, taskbar, Help menus, and Window Explorer; create directories and files; delete, rename and move files and directories

E. Number Systems
   1. Overview of base systems
   2. Base 10, 2, 8 and 16
   3. Conversion from one base to another

F. Programming Concepts
   1. History of Computation and programming languages
   2. Types and purposes of programming languages

G. Program Development
   1. The Notion of an Algorithm
   2. Top-Down design
   3. Pseudocode
   4. Flowcharts
   5. Design of correct, concise, and effective algorithms
   6. Control flow, basic logic, and Boolean expressions
   7. Usage of data types, arithmetic expressions, input/output, functions and arguments
   8. Effective methods for checking and debugging programs
   9. Programming in an integrated development environment (IDE)

VI. METHODS OF INSTRUCTION:

A. **Lecture** -
B. **Lab** - Hands-on exercises in the laboratory
C. **Demonstration** -
D. **Discussion** -

VII. TYPICAL ASSIGNMENTS:

A. Convert a series of decimal numbers into their binary, octal, and hexadecimal equivalents. B. Using pseudocode and flow charts, create the programming logic that will allow a user to enter a starting and ending value and display the even integer values (display the odd integer values). C. Using pseudocode and flow charts, creat the logic that will read in records from a sequential data file and process the records (emphasize that the file has an “unknown” number of records and what type of loop you would use).

VIII. EVALUATION:

A. **Methods**

B. **Frequency**

1. Frequency
   a. At least two in-class midterm examinations, or one in-class midterm examination and several quizzes
   b. Additional midterm examination(s) and/or quizzes
   c. In-class comprehensive final examination
   d. Programming assignments to cover each topic within the course content (contents can be combined). At least one of the programming assignments must be in-class, or covered by in-class examinations.
   e. It is suggested that there be at least one homework assignment each week to cover that week’s discussion. This might be written or a programming assignment.
2. Types of examination questions
   a. Write a program that incorporates and tests a function smallerOf() that returns the smaller of two given integer values.
   b. Write a program that inputs two integers from a file and outputs their sum and difference.
   c. Concern the decimal number 231 to hexadecimal and octal.
   d. Describe the fundamental operations of the Von Neumann computer architecture.
3. Evaluation Percentage Recommendations
   a. In-class grading percentage: 65%
      1. (In-class percentage may be increased, but not decreased.)
   b. For example:
      1. In-class programming assignments: 10%
      2. (May be included in the midterm and final exams)
      3. In-class midterm examination(s): 20%
      4. In-class final examination: 35%
      5. Other: 35% (e.g., programming assignments, chapter quizzes, written assignments, student presentations, etc.)

Note: It is suggested that programming assignments be of substantial difficulty to verify mastery of the subject area.

IX. TYPICAL TEXTS:

1. Venit, Stewart *Short Prelude to Programming: Concepts and Designs*., Scott/Jones Publishing, 2001.
2. Reed, David *A Balanced Introduction to Computer Science*., Pearson Prentice Hall, 2005.
3. Dale, Nell and John Lewis *Computer Science Illuminated*., Jones and Bartlett Publishing, 2002.
4. Farrell, Joyce *Programming Logic and Design: Comprehensive.* 2nd ed., Course Technology, 2004.
5. Sawyer and Williams *Using Information Technology.* 5th ed., McGraw-Hill/Irwin, 2002.

X. OTHER MATERIALS REQUIRED OF STUDENTS: