

Course Description

"C" Programming Language

I. Course Description

This course is designed to introduce computer programming fundamentals and methodologies using the "C" programming language. The student will gain general expertise in algorithm development, source code design, and program debugging, as well as particular knowledge and experience with the "C" programming language.

II. Instructional Material

Text: "Teach Yourself C in 24 Hours", Tony Zhang, SAMS Publishing, 1997

Compiler: Borland TurboC, Version 3.0

Computers: IBM-compatible personal computers with DOS, Windows 3.1, Windows 95, or Windows 98

III. Course Goals and Objectives

A. Goals

1. To expose the student to the fundamental concepts of computer programming
2. To develop expertise in the use of the "C" programming language
3. To develop an understanding of algorithm development and debugging
4. To gain experience in programming for real-world applications
5. To foster creativity in design and development of computer programs
6. To prepare students for further studies in "C++" or other computer languages

B. Objectives

1. The student will be able to develop programs in the "C" language
2. The student will be able to enter the source code, compile, link, and run the application
3. The student will be able to apply debugging logic to repairing faulty programs
4. The student will be able to utilize "C" variables, operators, and flow controls
5. The student will gain an understanding and working knowledge of pointers, arrays, and structures
6. The student will be able to sensibly utilize program flow control in "C" programs
7. The student will be able to utilize file, keyboard, and screen input and output in programs

IV. Course Outline

1. The basics of "C"
 - Your first "C" program - Hello, World
 - The essential of a "C" program
 - Data types and names
 - Reading and writing standard input/output
2. Operators and Flow Control
 - Manipulating data with operators
 - Repeating operations
 - More operators
 - Data modifiers and math functions
 - Conditional statements
3. Pointers and Arrays

- Introduction to pointers Storing similar data items Manipulating strings (arrays of char) Scope and storage classes
4. Functions, Dynamic Memory Allocation Functions in "C" Applying pointers Allocating memory More data type and functions
 5. Structures, unions, and file input/output
 - Collecting data items of different types
 - Unions
 - File input and output - Part 1
 - File input and output - Part 2
 - The "C" preprocessor
 - Where can we go from here?

V. Teaching Methods

- A. Teaching shall be broken into two distinct class times: lecture and lab. The lecture time shall be used to explain programming concepts and cover the material in the textbook. It shall also be used to provide examples of the concepts provided and allow student interaction in the practice of developing logic structures and algorithms. Lecture times shall include question an answer sessions.
- B. The lab shall be used to allow students to practice what they have learned in the lecture time and work on assignments and projects. Lab time shall include working through examples provided by the textbook and the teacher, as well as development of new works. The teacher or a lab assistant shall supervise lab sessions.

VI. Evaluation

- A. Components
 1. Final exams
 2. Chapter tests
 3. Homework assignments
 4. Semester projects
 5. Teacher-student interaction
- B. Grading scale
 - A 90-100%
 - B 80-89%
 - C 70-79%
 - D 60-69%
 - F 0-59%