

Course Description: This course provides students with a comprehensive study of the C programming language. Classroom lectures stress the strengths of C, which provide programmers with the means of writing efficient, maintainable, and portable code. The lectures are supplemented with non-trivial lab exercises.

Who Should Attend: This course is for programmers who have had experience in any programming language or have been tasked with a C programming project, and other technical types including managers and customer support engineers who need to know C.

Prerequisites: Experience with a programming or an assembly language is required.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Write C programs that are non-trivial.
- Use the variety of data types appropriate to specific programming problems.
- Utilize the modular features of the language.
- Demonstrate efficiency and readability.
- Demonstrate the use of the various control flow constructs.
- Use arrays as part of the software solution.
- Utilize pointers to efficiently solve problems.
- Include the structure data type as part of the solution.
- Create their own data types.
- Use functions from the portable C library.

Course Outline:

Getting Started

What is C?
Background
Sample Program
Components of a C Program
Examples
Data Types
Variables
Naming Conventions for C Variables
Printing and Initializing Variables
Array Examples
Compiling and Executing a C Program

Functions and Operators

Examples of C Functions
Functions
sum Invoked from main
Invoking Functions
Elementary Operators
The operator= Operators
Operators
The Conditional Operator
Increment and Decrement Examples
Increment and Decrement Operators

Control Flow Constructs

Examples of Expressions
if
if else
while
for
Endless Loops
do while
break and continue
switch
else if

The C Preprocessor

#define
Macros
#include
Conditional Compilation
#ifdef
#ifndef

Simple I/O

Character I/O
End of File
Simple I/O Examples
Simple I/O Redirection
I/O with Character Arrays

More on Functions

General
Function Declarations

Returning a Value or Not
Function Prototypes
Arguments and Parameters
Organization of C Source Files
Extended Example
The getline Function
The strcmp Function
The check Function
The atoi Function
The average Function
Summary

Bit Manipulation

Defining the Problem Space
A Programming Example
Bit Wise Operators
Bit Manipulation Functions
Circular Shifts

Strings

Fundamental Concepts
Aggregate Operations
String Functions

Higher Dimensional Arrays

Array Dimensions
An Array as an Argument to a Function
String Arrays

Separate Compilation

Compiling Over Several Files
Function Scope
File Scope
Program Scope
Local static
register and extern
Object Files
Libraries
The C Loader
Header Files

Pointers (Part 1)

Fundamental Concepts
Pointer Operators and Operations
Changing an Argument with a Function Call
Pointer Arithmetic
Array Traversal
String Functions with Pointers
Pointer Difference
Prototypes for String Parameters
Relationship Between an Array and a Pointer
The Pointer Notation *p++

Pointers (Part 2)

Dynamic Storage Allocation - malloc
Functions Returning a Pointer
Initialization of Pointers
gets - a Function Returning a Pointer
An Array of Character Pointers
Two Dimensional Arrays vs. Array of Pointers
Command Line Arguments
Pointers to Pointers
Practice with Pointers
Function Pointers

Structures

Fundamental Concepts
Describing a Structure
Creating Structures
Operations on Structures
Functions Returning Structures
Passing Structures to Functions
Pointers to Structures
Array of Structures
Functions Returning a Pointer to a Structure

Structure Related Items

typedef - New Name for an Existing Type
Bit Fields
unions
Non-Homogeneous Arrays
Enumerations

File I/O

System Calls vs. Library Calls
Opening Disk Files
fopen
I/O Library Functions
Copying a File
Character Input vs. Line Input
scanf
printf
fclose
Servicing Errors - errno.h
feof

Information About Files

The stat Function
File Existence
Telling Time - time and ctime
Telling Time - localtime

I/O With Structures

A Database Application
The menu Function
The fwrite Function
The create_db Function
The fread Function

The print_db Function
fseek
The retrieve_db Function
fflush and ftell

Useful Library Functions

strstr
strchr, strchr
system
strtok
strspn, strcspn
Math Functions
Character Testing Functions
exit and atexit
signal
memcpy and memset
qsort
Binary Search - bsearch

Appendix A: C Language Programming

Important Header Files
printf Formats
C Reserved Words
Conversion
Precedence Chart