The C Programming Language was originally created to write the UNIX operating system. It quickly turned into a multi-purpose language used by all types of programmers for a wide variety of applications. C is a small language that can be learned quickly. It is highly-structured and modular, supporting both small and large programs equally well.

Batky-Howell's C Programming course has been introducing students to the power and flexibility of this language for years. You will develop the ability to design and write programs in the ANSI Standard C programming language. Concepts such as arrays, functions, control flow, pointers, and many others will quickly prepare you to successfully code your own C applications.

Course Objectives:

- Describe the basic elements of C.
- Write C programs using all the major features of the language.
- Define and use C datatypes.
- Write variable declarations for programs.
- Apply the unique notations that C uses for assignments, incrementing, and decrementing.
- Control the flow of program execution.
- Write modular programs consisting of functions.
- Describe the purpose and functioning of a preprocessor.
- Define the relationship between arrays and pointers.
- Use structure variables for data storage and manipulation.

Audience: Programmers new to the ANSI C language.

Prerequisites: Programming skill in a language such as Pascal, COBOL, BASIC, or assembler.

Number of Days: 5 days

1. Course Introduction
- Course Objectives
- Overview
- Suggested References

2. Introduction to C
- What is C ?
- Features of C
- Why Program in C ?
- History of C
- Current Status and Future

3. An Overview of C
- The First Program (hello.c)
- How to Compile and Run a C Program
- An Arithmetic Program (roof.c)
- Execution Flow Control (mph.c)
- The for Loop
- The for Loop - Diagram
- Character I/O
- A File Copier Program (cp2.c)
- A Character Counter (wc2.c)
- A Look at Arrays
- Stock Values (stock1.c)
- The char Data Type
- Strings (Character Arrays)
- A String Copy Program (stringcp.c)
- A Look at Functions
- A Functional Program (func1.c)
- A Review of printf()

4. Data Types and Variables
- Fundamental Data Types
- Data Type Values and Sizes