Price: R12,900.00 excl. VAT
Duration: 5 days
Code: SANSC

Standard C Programming

Description

Learning C is one of the best foundations for any programmer. Many of the principles in C are used in other languages, and C will give you a better understanding of what happens when your code runs. This course will teach you the fundamentals of the C language so that you can write C programs, or maintain existing C code. You will learn to write portable code that can be used on different platforms and devices. This course is also an essential foundation for anyone who wants to learn C++.

Objectives

After you have completed the Standard C Programming course, you will be able to:

- Write your own C programs for any operating system.
- Read and maintain C programs.
- Debug C code to find and correct mistakes.
- Understand the C compiler and the pre-processor.
- Use pre-processor directives.
- Understand the importance of portable code and standards compliance.

Intended Audience

You should attend the Standard C Programming course if:

- You are a programmer and you want to learn the C language.
- You need to support existing C systems.
- You want to learn C++ but do not yet know C.
- You have already learnt some C - perhaps on your own or at university - but struggle with some concepts or have gaps in your knowledge.
- You need to use C for embedded systems.

Prerequisites

Before you attend the Standard C Programming course:

- You must already be a programmer and have experience in programming.

Course Contents

Introduction

- Overview of the C language.
- Portability.
- Terminology.
- Programming fundamentals.
- Structured programming principles.

The C Compiler

- Compiler operation in general.
- The preprocessor.
- Output results of the compilation phases.
- Object files and libraries.
- Installing a C compiler.

- Creating program files.
- Compiler code generation, the linking process and executing programs.
- The C startup module and the main function.
- Separate compilation.

C Language Fundamentals

- Keywords.
- Program structure and conventions.
- The standard library.
- Header files.
- Functions and I/O functions.
- Character set.
- Literal values.
- Comments.
- Preprocessor directives.
- Fundamental types, derived types, structured types, enumerated types, user defined types.
- Storage classes.

Expressions and Operators

- Expressions and expression results.
- Operators in expressions.
- Type requirements of operators.
- Implicit and inherent type conversions.
- Types of operands and the resulting types.
- Bitwise, relational, logical, compound assignment and other operators.

Functions and Statements

- Function definitions, declarations and prototypes.
- Passing parameters.
- Recursive functions.
- Function returns.
- The function call operator.
- Local variables.
- Function pointers and the pointers-to-function data type.
- Types of statements.

Pointers, Arrays and Structs

- Pointer types and operators.
- Pointer arithmetic.
- Array subscripts.
- Indirection.
- Multi-dimensional arrays.
- Pointer-to-array and pointer-to-function types.
- Structs and unions.
- Member selection, indirect member selection.

*** The lecturer reserves the right to modify the contents of the course to suit the needs of the delegates.*