

Price: R9,800.00 excl. VAT
Duration: 4.5 days
Code: AANSC

Advanced C Programming

Description

This course will take you to the next level as a C programmer. You will learn advanced C programming concepts, and techniques to make your code more efficient and more portable.

Objectives

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

- Understand advanced C programming techniques.
- Understand and use all types of pointers.
- Use self-referencing structures and link lists.
- Manage memory within C applications.
- Understand performance-related issues.

Intended Audience

You should attend the Advanced C Programming course if:

- You are a C programmer and you want to learn about the advanced aspects of the C language.
- You are a C programmer and you want to learn to write better C code.
- You are an engineer working with embedded systems written in C.

Prerequisites

Before you attend the Advanced C Programming course:

- You must have attended our Standard C Programming course or already be comfortable with the fundamentals of the C programming language.
- You should have at least 6 months practical experience programming in C.

Course Contents

Introduction

- Professionalism.
- Portability.
- Modularity.
- Structured programming.
- Coding standards.
- Standard library functions.
- Non-standard functions.
- Under-utilized functions.

The C compiler

- Technical aspects.
- Startup module.
- Modifications for embedded / custom systems.
- Translation order and tokenization.
- Object modules.
- Compiler switches.

The C Preprocessor

- Conditional compilation.
- Token concatenation.
- Stringization.
- Charization.
- Compilation units.
- Trigraphs.
- Advanced macros.

Expressions, Types and Variables

- Expression evaluation.
- Operator association and precedence.
- Sequence guarantee points.
- Implicit/explicit type conversions.
- Literals.
- Escape characters.
- Type hierarchy.
- Derived types.
- Bitfields.
- Portability.

Functions, Statements and Program Structure

- Overview.
- Stack usage for local variables and parameters.
- Function returns.
- Function call operator.
- Variable number of parameters.
- Structured programming vs OOP techniques.
- Libraries and make files.

Arrays and Pointers

- Array names.
- Array manipulation.
- Multi-dimensional arrays.
- Pointer variables.
- Indirection.
- Dynamic arrays and pointers.
- Pointers to structures and functions.

Memory management

- The C runtime memory.
- Stack, heap and static data areas.
- Dynamic memory.
- Fragmentation of memory.
- Memory management techniques.

Data Structures

- Creating and using structures.
- Self-referencing structures.
- Linked lists.
- Generic data structures.
- Portability.
- Memory management.
- Information hiding.

Debugging Techniques

- Stabilization: lexical, syntactic, execution, logic errors.
- Error proximity.
- Debuggers.
- Memory initialization.
- Structure sentinels.
- Stack errors.
- Conditional compilation.

Miscellaneous

- Introduction to numerical methods, encryption, sorting and algorithm evaluation.
- Introduction to C++.
- Graphical User Interfaces.

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