

**Price:** R11,900.00 excl. VAT  
**Duration:** 5 days  
**Code:** SCPLS

# Standard C++ Programming

## Description

C++ is one of the most powerful languages for serious systems that need real speed, high performance and high reliability. This course will teach you the fundamentals of the C++ language, in line with the newest ISO standards. You will learn how to write C++ code that can be used on different platforms. You learn to read, debug and maintain C++ code, and how object-oriented concepts apply to the C++ language.

## 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 principles of object orientation as used in C++.
- Understand the importance of portable code and standards compliance.
- Understand generic programming and the use of templates in C++.

## Intended Audience

You should attend the Standard C++ Programming course if:

- You are a C programmer and you want to learn the C++ language.
- You need to support existing C++ systems.
- You have been using C++, 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 have attended our Standard C Programming 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

### ***A Different and Enhanced C***

- Const types.
- Character literals and char arrays.
- Structured and enumerated types.
- Empty parameter lists.
- Void pointers.
- C++ comments.
- Type checking.
- Operators as functions.
- Default parameters.
- Inline functions.
- References.

- Dynamic memory allocation.
- Exception handling.
- Scope resolution.
- Namespaces.

### ***Object Oriented Programming Basics***

- Data encapsulation.
- Structured programming and OOP.
- Classes and objects.
- Inheritance and abstract data types.
- Polymorphism.

### ***C++ Standard Library Overview***

- Streams and overloaded stream operators.
- I/O Manipulators.
- Strings and the STL.
- Collections and iterators.
- Generic algorithms.

### ***Classes and Objects***

- Member functions, constructors and destructors.
- Constructor overloading.
- Copy constructor.
- Canonical form for classes.
- Overloaded assignment operator.
- Shallow versus deep copies.
- Dynamic memory allocation in constructors.

### ***Data Members***

- Structured type members.
- Access specifiers.
- Static members.

### ***Expressions and Functions***

- Expression evaluation.
- C++ operators and precedence.
- Implicit type conversion.
- Function signatures.
- Overloading functions.
- Variable argument lists.
- Pass and return by references.

### ***Templates and the STL***

- Templates and generic functions.
- Deriving from template classes.

### ***Friends, Operators and Member Functions***

- Friend functions.
- Overloading standard operators.

- The this keyword.
- Inline member functions.
- Scope resolution.
- Static functions.

### ***Inheritance***

- Base and derived classes.
- Initializer lists.
- Public, private and protected inheritance.
- Polymorphism in detail.
- Multiple inheritance.
- Virtual base classes.

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