IMPACTFUL
Powered by LRMG

QCTO
Quality Council for Trades & Occupations

IT SKILLS PROGRAMMES

C++ PROGRAMMER

QUALIFICATION TYPE:
SKILLS PROGRAMME ID (SP-230374)

TRAINING DAYS: 75

Designed to support your Workplace Skills Plan and build job-ready IT capability

Build high-performance coding capability for real-world systems

Efficient, reliable software underpins critical systems across industries. Organisations need developers who can build solutions that perform under pressure.

This C++ Programmer skills programme builds practical capability to develop, test, and optimise software with performance in mind. Learners strengthen their ability to write structured code, solve problems logically, and build fast, reliable applications.

Whether you are strengthening your programming foundation or looking to grow into more advanced development roles, this programme supports your progression into performance-focused software development.

On successful completion of this skills programme and successful FISA (Final Integrated Summative Assessment), learners will be awarded: **QCTO Certificate: C++ Programmer**
(An accredited, credit-bearing Skills Programme certificate)

We work closely with you to understand your objectives, guide you through the requirements, and support the implementation of skills programmes that deliver real impact.

NQF LEVEL 4

CREDITS 60

WHO SHOULD ENROL?

- Organisations strengthening software development and technical coding capabilities
- Teams working on performance-critical systems and applications
- Individuals with a foundation in programming looking to specialise in C++
- Learners interested in building strong software development skills
- Professionals wanting to strengthen their technical coding capability

WHAT MAKES THIS COURSE DIFFERENT?

This skills programme is designed to move you from learning to doing

- Build strong foundations in structured programming and problem-solving
- Apply your skills through hands-on coding and development tasks
- Gain experience working with real-world, performance-focused scenarios

WHAT IS THE ENTRY CRITERIA?

- NQF Level 4

IMPACTFUL
Powered by LRMG

QCTO
Quality Council for Trades & Occupations

C++ PROGRAMMER

SKILLS PROGRAMME

The purpose of the C++ Programmer skills programme is to equip learners with the competence to apply programming principles, object-oriented programming concepts, intermediate C++ functionalities, version control skills, and software development life cycle processes to develop efficient, functional and readable software solutions. The skills programme prepares learners for employment or career progression in software development, robotics, embedded systems, game development, and IT support roles.

This skills programme is designed to equip learners with practical and industry relevant C++ programming capabilities required in the modern digital and software development environment. The skills programme addresses the growing demand for developers who can apply programming principles, object-oriented design, debugging techniques, and intermediate C++ functionalities to build efficient, scalable and reliable software solutions.
By applying methodical and procedural programming techniques, learners gain the ability to analyse problems, design solutions, implement C++ programs, and troubleshoot issues within real world development environments.

Skills your teams will build
Upon successful completion of the skills programme, learners are able to develop, implement and troubleshoot basic to intermediate C++ programs by applying the principles of programming, object-oriented programming, version control, and software development life-cycle practices to produce functional, efficient and readable software solutions within a collaborative environment

Assessment designed to show what you can do

Learners are assessed throughout the programme using a variety of methods, which may include practical tasks, written assignments, short projects, demonstrations, and presentations. Evidence of learning is collected and recorded for monitoring, feedback, and quality assurance. Where the curriculum is delivered in modules, internal summative assessments are conducted at the end of each module and results are recorded. After completing all modules, learners must complete a Final Integrated Supervised Assessment (FISA) that integrates the key outcomes of the skills programme. The FISA is implemented through one assessment process, which may be conducted using either of the following supervised methods:

Face-to-face Assessment

The FISA is conducted in person under direct supervision, using approved assessment instruments and a rubric and/or checklist to confirm that all required competencies have been demonstrated

Virtual delivery via e-assessment

The FISA may be conducted virtually via our secure e-assessment platform (Questionmark). This assessment is conducted under supervised conditions and is further strengthened through the use of proctoring, which provides real-time monitoring and verification of learner identity and assessment conditions. Proctoring enhances the integrity, credibility, and reliability of the FISA by reducing the risk of malpractice, ensuring compliance with assessment rules, and confirming that the assessment is conducted fairly, consistently, and in line with approved assessment requirements

The FISA is supervised, with a pass mark set at 75%

Let's partner for impact!

Our approach combines a deep understanding of your objectives with expert guidance on QCTO skills programmes, ensuring smooth implementation and meaningful impact in the workplace.

We'll help you get clear on the holistic implementation process
From first conversation to final assessment, you'll be supported by a team that understands how to make QCTO programmes work in practice.

Delivered your way
- Classroom | Johannesburg
- Virtual | Instructor-led
- On-site | Nationwide

Take the next step with us!

impactul@lrmg.co.za
impactful.co.za

C++ PROGRAMMER

QUALIFICATION TYPE:
SKILLS PROGRAMME ID (SP-230374)

This detailed overview outlines how the skills programme is structured to develop capability progressively, from foundational knowledge, through applied practical skills, to integrated workplace experience. Each module is aligned to the credit requirements of the nationally recognised skills programme

The Skills Rationale

This skills programme is designed to equip learners with practical and industry relevant C++ programming capabilities required in the modern digital and software development environment.

The skills programme addresses the growing demand for developers who can apply programming principles, object oriented design, debugging techniques, and intermediate C++ functionalities to build efficient, scalable and reliable software solutions.

KNOWLEDGE COMPONENTS

Module 1: C++ Basics

- 1.1 Basic computer knowledge
- 1.2 Basic concepts of C++
- 1.3 Introduction to suitable IDE (Integrated Development Environment)
- 1.4 GIT and GitHub (Global Information Tracker)
- 1.5 Problem solving in programming
- 1.6 Life cycle for developing a solution
- 1.7 Five basic Concepts of C++
- 1.8 Fundamental concepts in C++
- 1.9 C++ syntax

Module 2: Principles of Programming with C++

- 2.1 Variables
- 2.2 C++ Strings
- 2.3 Operators
- 2.4 Conditions in C++
- 2.5 Switch statements in C++
- 2.6 Arrays in C++
- 2.7 Loops in C++
- 2.8 Reference and enumerations in C++
- 2.9 Exception handling in C++
- 2.10 Dynamic Arrays in C++
- 2.11 Pointers in C++
- 2.12 C++ Char Data Types
- 2.13 File handling in C++: Basic input/output
- 2.14 C++ structure (Struct)
- 2.15 C++ class and object
- 2.16 C++ operator overloading
- 2.17 STD: list in C++
- 2.18 C++ Functions
- Date/time in C++
- Debugging in C++

Module 3: Principles of Object orientated Programming with C++

- 31. Object-Oriented Programming (OOP in C++)
- 3.2 Object-Oriented Programming in C++ concepts
- 3.3 OOP Capabilities

Module 4: Principles of intermediate programming with C++

- 4.1 Structures and Classes
- 4.2 Polymorphism in C++
- 4.3 Templates
- 4.4 Namespaces
- 4.5 Vector in STL (Standard Template Library)
- 4.6 Stacks in C++ STL
- 4.7 Map in C++ STL
- 4.8 Algorithms in C++
- 4.9 Mini Databa in C++

APPLICATION COMPONENTS

Module 1: Getting started with C++

- 1.1 Basic computer Skills
- 1.2 C++ Basics: download, install, run and create first C++ programme
- 1.3 Getting started with an IDE
- 1.4 Use GIT commands and use GITHub Functionalities
- 1.5 Apply the programming life cycle to develop a solution
- 1.6 Create and run a C++ class

Module 2: Programming with C++

- 2.1 Declare variables in C++
- 2.2 Declare strings in C++
- 2.3 Use operators in C++
- 2.4 Use Conditions
- 2.5 Use switch statements in C++
- 2.6 Declare and initialise arrays in C++
- 2.7 Use loops in C++
- 2.8 Create and declare enumerations and references in C ++
- 2.9 Use exception handling functions to find and mitigate exceptions
- 2.10 Create, initialise and delete dynamic arrays
- 2.11 Use pointers in C++
- 2.12 Declare a char (character)
- 2.13 File handling in C++
- 2.14 Use C++ Struct
- 2.15 Create a class and an object
- 2.16 Overload an Operator
- 2.17 Use Std: list in C++
- 2.18 Write and call a function in C++
- 2.19 Use date and time in C++
- 2.20 Debug code

Module 3: Object oriented programming with C++

- 3.1 Use objects, properties, functions and methods and create objects in Object-Oriented Programming in C++
- 3.2 Use OOP capabilities in programming

Module 4: Intermediate Programming with C+

- 4.1 Structures and classes
- 4.2 Call an overloaded function and operator
- 4.3 Create templates
- 4.4 Namespace declaration
- 4.5 Vectors in STL
- 4.6 Implement stacks in STL
- 4.7 Create a map in C++ STL
- 4.8 Algorithms
- 4.9 Create a mini database

Module 5: Projects with C++

- 5.1 C++ Exercises may be conducted at the end of the skills programme or may be integrated into learning at the end of the applicable module
- 5.2 Use a known framework (of own preference) with functions to build a game in C++
- 5.3 Use a known framework (of own preference) with functions to assemble a complete solution applying the below aspects given contextualised context within a specific sector