

Price: R5,900.00 excl. VAT
Duration: 5 days
Code: INTRO

Introduction to Programming

Description

The Introduction to Programming course is designed for delegates who have no programming experience. It covers the fundamental concepts of programming: data types, variables, conditional and iterative statements, modular design and error handling. It also includes a brief introduction to object orientation and relational databases.

Objectives

Delegates who complete the Introduction to Programming course will be able to:

- Identify the components of a computer system.
- Understand different numbering systems.
- Understand the fundamental concepts of programming: data types, variables, constants, operators, iterative statements, conditional statements and procedures.
- Write small scripts.
- Understand the Software Development Life Cycle.

Intended Audience

Delegates with no programming experience who want to learn to program.

Project managers who want to have a better understanding of what programming entails.

Prerequisites

Delegates must be familiar with files and directories, and using a text editor.

Course Contents

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

Introduction to Computers • A brief history of computers. • The impact of computers on society. • Computer architecture. • Hardware components. • System software. • The role of the operating system. • Types of operating systems. • Types of application software.

Programming languages • The concept of a programming language. • Classification of programming languages. • Compilers and interpreters. • Source code and object code.

Numbering systems • Binary, octal, decimal and hexadecimal numbering systems. • Internal representation of data. • Binary arithmetic.

Building blocks of programming languages • Generally available language constructs.
• Variables and constants. • Local and global variables. • Operators and operator precedence.
• Data types. • Flow control structures: iterative and conditional statements. • Truth tables: AND, OR, XOR. • Subroutines and functions. • Parameters. • Nesting. • Debugging and error handling.

The Systems Development Life Cycle • The development life cycle. • Systems planning and analysis: user requirements. • Systems design and its importance. • Implementation and testing. • Support. • Software engineering models. • The importance of documentation and coding standards.

Other Topics • Structured programming. • Object orientation. • Relational databases.