COURSE OVERVIEW
The Foundation Level qualification is aimed at professionals who need to demonstrate practical knowledge of the fundamental concepts of software testing.

The ISTQB Foundation Certificate in Software Testing provides a competitive advantage to companies who want their software testing to match international standards.

COURSE CONTENT
The course covers the following areas:

FUNDAMENTALS OF TESTING
The course begins by defining testing terminology and all the fundamental aspects of the testing process such as defining what the process is and why testing is necessary. We will cover general testing principles, the fundamental test process and the psychology of testing. Important concepts in testing such as the difference between regression testing and retesting are discussed. There is an emphasis on strong communication and the importance of independence when it comes to successful testing.

TESTING THROUGHOUT THE SOFTWARE LIFE CYCLE
In this module we discuss the role of testing in the software development life cycle, the targets of testing, and we define maintenance testing. We will also cover software development models and what constitutes good testing in any model. Test levels are discussed within a V-model and a clear distinction is made between the roles of verification and validation testing. Important principles governing the economic viability of testing will also be addressed.
The Foundation Level qualification is aimed at professionals who need to demonstrate practical knowledge of the fundamental concepts of software testing. The ISTQB Foundation Certificate in Software Testing provides a competitive advantage to companies who want their software testing to match international standards.

**COURSE CONTENT**

- **FUNDAMENTALS OF TESTING**
  - The course begins by defining testing terminology and all the fundamental aspects of the testing process such as defining what the process is and why testing is necessary. We will cover general testing principles, the fundamental test process and the psychology of testing. Important concepts in testing such as the difference between regression testing and retesting are discussed. There is an emphasis on strong communication and the importance of independence when it comes to successful testing.

- **TESTING THROUGHOUT THE SOFTWARE LIFE CYCLE**
  - In this module we discuss the role of testing in the software development life cycle, the targets of testing, and we define maintenance testing. We will also cover software development models and what constitutes good testing in any model. Test levels are discussed within a V-model and a clear distinction is made between the roles of verification and validation testing. Important principles governing the economic viability of testing will also be addressed.

- **DYNAMIC TESTING TECHNIQUES**
  - Dynamic testing implies that testing is performed while the program code is running. The techniques are equivalence partitioning, boundary value analysis, decision logic tables, state transition testing, negative testing and random testing. Since the majority of testing is done in one of these techniques, these techniques have to be understood for the testing process to be effective. ‘Effective’ means that no serious faults are missed.

- **STATIC TECHNIQUES**
  - This module covers techniques performed without executing any program code. The two different test processes, using static or dynamic techniques, are distinguished. The static testing techniques covered are reviews, data flow analysis, control flow graphing, and complexity analysis. Static analysis through the use of tools can be beneficial, especially in discovering faults in areas of the code, which would typically be unreachable.

- **TEST DESIGN TECHNIQUES**
  - In this module the identification of test conditions and the design of test cases are addressed. Test design techniques: specification-based, structure-based and experience based. Examples of case testing are discussed, as well as factors to consider in the choice of techniques.

- **TEST MANAGEMENT**
  - This module covers important test management principles, such as, the benefits and pitfalls of various test organisation structures, influences on test planning and estimation, test progress monitoring and control with appropriate tools, and more. Independence, reporting and the use of appropriate tools are emphasised.

- **TOOL SUPPORT FOR TESTING**
  - In this module key issues for the effective use of tools are discussed. Specifically types of test tools, the effective use of the tools, including their benefits and risks, and the introduction of a tool into an organisation are the core issues. A summary of tool types and specific considerations are covered throughout the course.

**INTENDED AUDIENCE**

This ISTQB course will be of significant value to testers who have 3 months or more testing experience, test analysts, test managers who have not yet earned the qualification, business analysts, and developers.

**TRAINING VENUES**

iLAB Training presents courses at several well-established training facilities in Johannesburg, Cape Town and Durban. Training can be arranged at the client’s premises depending on adequate facilities and candidate numbers.

**COURSE DURATION**

An experienced software-testing practitioner who has been accredited by ISTQB presents the course. The course duration is three days. It runs from 8:00 to 17:30. This includes a lunch and two tea breaks daily.

**TRAINING STYLE**

The course is conducted in a classroom style. Candidates use practical tasks to consolidate theoretical concepts.

**TRAINING STYLE**

The course is conducted in a classroom style. Candidates use practical tasks to consolidate theoretical concepts.

**TESTING THROUGHOUT THE SOFTWARE LIFE CYCLE**

In this module we discuss the role of testing in the software development life cycle, the targets of testing, and we define maintenance testing. We will also cover software development models and what constitutes good testing in any model. Test levels are discussed within a V-model and a clear distinction is made between the roles of verification and validation testing. Important principles governing the economic viability of testing will also be addressed.

**DYNAMIC TESTING TECHNIQUES**

Dynamic testing implies that testing is performed while the program code is running. The techniques are equivalence partitioning, boundary value analysis, decision logic tables, state transition testing, negative testing and random testing. Since the majority of testing is done in one of these techniques, these techniques have to be understood for the testing process to be effective. ‘Effective’ means that no serious faults are missed.

**STATIC TECHNIQUES**

This module covers techniques performed without executing any program code. The two different test processes, using static or dynamic techniques, are distinguished. The static testing techniques covered are reviews, data flow analysis, control flow graphing, and complexity analysis. Static analysis through the use of tools can be beneficial, especially in discovering faults in areas of the code, which would typically be unreachable.

**TEST DESIGN TECHNIQUES**

In this module the identification of test conditions and the design of test cases are addressed. Test design techniques: specification-based, structure-based and experience based. Examples of case testing are discussed, as well as factors to consider in the choice of techniques.

**TEST MANAGEMENT**

This module covers important test management principles, such as, the benefits and pitfalls of various test organisation structures, influences on test planning and estimation, test progress monitoring and control with appropriate tools, and more. Independence, reporting and the use of appropriate tools are emphasised.

**TOOL SUPPORT FOR TESTING**

In this module key issues for the effective use of tools are discussed. Specifically types of test tools, the effective use of the tools, including their benefits and risks, and the introduction of a tool into an organisation are the core issues. A summary of tool types and specific considerations are covered throughout the course.

**COURSE PREREQUISITES**

Previous experience in software testing is not necessary; however, general knowledge of the information technology industry is an advantage.