

Certified Tester Foundation Level(CTFL)

ISTQB : International Software Testing Qualifications Board

Heading:

The **International Software Testing Qualifications Board (ISTQB)** is an internationally recognized non profitable software testing qualification certification association announced in 2002. ISTQB Certified Tester is a standardized qualification for software testers. ISTQB has different certifications, where Certified Tester Foundation Level(CTFL) is the one that should be taken first to take other certifications. CTFL is the one that most testers are taking and is demanded from job environments too.

What I am going to learn throughout this course? You will learn:

In this training you will be able to learn complete CTFL documents with practical and exam oriented examples. From the training you will get the following benefits.

ISTQB Exam Challenges	Our Solutions
Document is huge and difficult to cover and remember all.	We have prepared the huge ISTQB document into short and precise one so that anyone can cover it easily.
Uses same font and style and not easy to find out main points	We have bulleted out core points where most exam questions are focusing.
The exam uses a lot ambiguous terminologies which makes many people to fail the exam	We have screened out ambiguous terminologies and defined them clearly.
There are a lot ideas confuses you even if you read the document.	We cleared out confusing ideas by providing visual and practical samples so that you will not miss questions.
	We rate you areas and points where exam is focusing.
Lack of mock questions and explanations	Give you hundreds of questions with explanation which is dumped from past exams
No audio and video support	Give you audio based support

Our Exam Preparation Documents:

ISTQB has six syllabuses to prepare yourself to pass the CTFL exam. We have prepared these documents in such a way that you can understand it easily and highlighted main area where past exams were focussed.

- These six documents are:
 - Fundamentals of Software Testing
 - Testing Through the Software Life Cycle
 - Static Techniques
 - Test Design Techniques
 - Test Management
 - Tools to Support Testing
- Document content structure
 - Documents are shortened and bulleted very well to cover and identify main points easily
 - Documents are highlighted technical, special and confusing words with their definitions so that you can remember it easy
 - Documents are supported with audios and practical examples so that you can clear your confusion
 - Documents are clearly highlighted to show where most exam questions are focusing.

Mock Exams:

- We have 300 interview questions which are dumped from ISTQB past exams.
- Mock exams are subdivided into sections/syllabus so that it will be easy to remember.
- Mock exams are with answers and explained each in detail to answer why and why not questions

Exam Size and Time

- Exam has 40 multiple-choice questions, scoring of 1 point for each correct answer.
- Pass mark of 65% (26 or more points) a duration of 60 minutes (or 75 minutes for candidates taking exams that are not in their native or local language).

Exam questions are distributed across K-levels (Cognitive levels), which represent deepening levels of knowledge, as shown in the following table:

<i>Exam</i>	<i>Number of questions per K levels</i>				
	<i>K1(Remember)</i>	<i>K2(Understand)</i>	<i>K3(Apply)</i>	<i>K4(Analyze)</i>	<i>Total</i>
<i>Foundation</i>	<i>20</i>	<i>12</i>	<i>8</i>		<i>40</i>

Exam questions are distributed across Syllabus chapters as shown in the following table:

<i>Exam</i>	<i>Number of questions per Chapter</i>						
	<i>Ch. 1</i>	<i>Ch. 2</i>	<i>Ch. 3</i>	<i>Ch. 4</i>	<i>Ch. 5</i>	<i>Ch. 6</i>	<i>Total</i>
<i>Foundation</i>	<i>7</i>	<i>6</i>	<i>3</i>	<i>12</i>	<i>8</i>	<i>4</i>	<i>40</i>

Exams may be taken as part of a course delivered by an [Accredited Training Provider](#) or taken independently at an ex

Where to Register and Take Exam:

- ISTQB exam should be taken physically. See the list of [exam center locations](#).
- When you are ready to register, [create an account on ASTQB Exam Registration site](#).
- Once your account is created you can login and:
 - choose your exam from the catalog
 - choose the center you want to test at.
 - choose the time you want to start.
- If you just want to see where you can take the test and when, simply select the exam and the center but don't proceed to the checkout until you are ready.

Exam Prerequisite

- **Certified Tester Foundation Level (CTFL) exam** : no prerequisite

Canceling or Rescheduling Penalty

- Select your date and time for the exams carefully!
- There is a \$75 fee for a reschedule or cancellation within 72 hours of your exam.

For more information please check the curriculum...

Audience and Requirements for our Training:

- QA, BA and anyone who is interested to know and apply automation Testing.
- Needs prior knowledge about software testing fundamentals. Check “Fundamentals of Software Testing” from our portal for more details.

Training Highlights:

- Mentorship will be available through training and after training
- You will be look different document as a simple and do samples
- You will practise on interview questions and secession from day one

Outline

ISTQB - Foundation Level Short Course Outline					
Fundamental of Testing	Testing Throughout the Software Life Cycle	Static Techniques	Test Design Techniques	Test Management	Tools Support for Testing
Why is testing necessary?	Software development models	Static techniques and the test	The test development process	Test organization	Types of test tools

		process			
What is testing?	Test levels	Review process	Categories of test design process	Test planning and estimation	Effective use of tools: Potential benefit and risks
Seven testing principles?	Test types	Static analysis by tools	Specification based techniques(Black-Box)	Test process monitoring and control	Introducing a tool to an organization
Fundamental test process	Maintenance Testing		Structure based techniques (White box)	Configuration management	
The psychology of testing			Experienced technique	Risk and testing	
Code of Ethics			Choosing test techniques	Incident management	

Fundamentals of Software Testing

1. Software testing
 - a. Necessity
 - b. Objective and purpose
2. Software testing principles
 - a. Testing is to show presence of defect
 - b. Early testing is better
 - c. Exhaustive testing is impossible
 - d. Defect clustering is ideal to solve exhaustive testing
 - e. Pesticide paradox
 - f. Testing is context dependent
 - g. Absence of errors fallacy
3. Software incidents
 - a. Error
 - b. Defect / Bugs
 - c. Faults
 - d. Failure
4. About defects and failures
 - a. Source of defects and failures

- b. Cost of defects
- c. Defect life cycle
- d. Defect severity and priority?
- 5. Fundamental test process (testing life cycle)
 - a. Planning and Control
 - b. Analysis and Design
 - c. Implementation and Execution
 - d. Evaluating exit criteria and Reporting
 - e. Test Closure activities
- 6. Testing independence
 - a. Test by own person
 - b. By another person but inside the team
 - c. By internal testing team
 - d. Outsourced test
- 7. Software Quality
 - a. Good design
 - b. Good functionality
 - c. Reliable
 - d. Consistency
 - e. Durable
 - f. Good after sales service
 - g. Value for money

Testing throughout the testing lifecycle

- 1. Verification
 - a. Review
 - b. Walkthrough
 - c. Inspect
 - d. Advantage of verification
- 2. Validation
- 3. Capability Maturity Model (CMM-Levels)
 - a. Purpose
 - b. Initial
 - c. Managed
 - d. Defined
 - e. Quantitatively Managed
 - f. Optimizing
- 4. Software Development Life Cycle(SDLC)
 - a. Requirement gathering and analysis
 - b. Design
 - c. Implementation or coding
 - d. Testing
 - e. Deployment

- f. Maintenance
- 5. Software Development Models
 - a. Waterfall model
 - b. V-model
 - c. Incremental model
 - d. RAD model
 - e. Agile model
 - f. Iterative model
 - g. Spiral model
 - h. Prototype Model
- 6. Software Testing Levels
 - a. Unit testing
 - b. Component testing
 - c. Integration testing
 - i. Big bang integration testing
 - ii. Top down
 - iii. Bottom up
 - d. Functional incremental
 - e. Incremental testing
 - f. Component integration testing
 - g. System integration testing
 - h. System testing
 - i. Acceptance testing
 - j. Alpha testing
 - k. Beta testing
- 7. Software Test Types
 - a. Functional testing
 - b. Non-functional testing
 - c. Structural testing
 - d. Testing related to changes
- 8. Reliability testing
 - a. Modeling
 - b. Measurement
 - c. Improvement
- 9. Usability testing
 - a. Learnability
 - b. Memorability
 - c. Error recovery
 - d. Satisfactory
- 10. Efficiency testing
- 11. Maintainability testing
- 12. Portability testing
- 13. Baseline testing

14. Compliance testing
15. Documentation testing
16. Scalability testing
17. Performance testing :
 - a. load, stress,
 - b. endurance, volume, spike
18. Security testing
19. Recovery testing
20. Internationalization testing and Localization testing
21. Confirmation testing
22. Regression testing
23. Structural testing
24. Maintenance Testing
25. Impact analysis

Static Techniques

1. Test design techniques
 - a. Static Techniques
 - b. Dynamic Techniques
2. Static testing reviews
 - a. Walkthrough/informal
 - b. Technical review/ less formal
 - c. Inspection/formal
3. Formal review processes
 - a. Planning
 - b. Kick-off
 - c. Preparation
 - d. Review meeting
 - e. Rework
 - f. Follow-up
4. Formal review roles and responsibilities
 - a. Moderator
 - b. Author
 - c. Scribe
 - d. Reviewers
 - e. Managers

Test Design Techniques

1. Test analysis
2. Traceability
3. Test implementation
4. Test design technique

5. Black box testing or Specification-based
 - a. Equivalence partitioning (EP)
 - b. Boundary Value Analysis (BVA)
 - c. Decision tables
 - d. State transition testing
 - e. Use case testing
6. White box testing or Structure-based
 - a. Experience-based testing
 - i. Error guessing
 - ii. Exploratory testing
 - b. Structure based technique
 - c. Test coverage
 - i. Why we do test coverage?
 - ii. When do we apply it?
 - iii. Statement coverage
 - iv. Branch or decision coverage
 - v. Condition coverage

Test management

1. Roles and responsibilities of a Test Leader
 - a. Prepare test plan and test strategies
 - b. Monitor
 - c. Control
 - d. Estimate the testing
2. Roles and responsibilities of a Tester
 - a. Participate during test planning
 - b. Design test case
 - c. Review any document developed in STLC
 - d. Report defects
3. IEEE 829 test plan standard: purpose
 - a. Test plan identifier
 - b. Test deliverables
 - c. Introduction
 - d. Test tasks
 - e. Test items
 - f. Environmental needs
 - g. Features to be tested
 - h. Responsibilities
 - i. Features not to be tested
 - j. Staffing and training needs
 - k. Approach Schedule
 - l. Item pass/fail criteria

- m. Risks and contingencies
 - n. Suspension and resumption criteria
- 4. Things keep in mind while planning
- 5. Estimation techniques
- 6. Factors affecting test effort
- 7. Test strategy
- 8. Test monitoring
- 9. Test control
- 10. Configuration management
- 11. Risks in software testing
 - a. Product risk
 - b. Project risk
 - c. Risk-based testing
 - d. Risk analysis
- 12. Incident management and logging
 - a. What are incident reports?
 - b. How to write a good incident report?
 - c. What is test status report?

Tool support for testing

- 1. Tool support for management of testing and tests:
 - a. Test management tools
 - b. Requirements management tools
 - c. Incident management tools
 - d. Configuration management tools
- 2. Tool support for static testing:
 - a. Review process support tools
 - b. Static analysis tools (D)
 - c. Modelling tools (D)
- 3. Tool support for test specification:
 - a. Test design tools
 - b. Test data preparation tools
- 4. Tool support for test execution and logging:
 - a. Test execution tools
 - b. Test harness/ Unit test framework tools (D)
 - c. Test comparators
 - d. Coverage measurement tools (D)
 - e. Security tools
- 5. Tools support for performance and monitoring:
 - a. Dynamic analysis tools (D)
 - b. Performance testing, Load testing and stress-testing tools
 - c. Monitoring tools

