

## Introduction to Programming and Databases for Testers Course Outline

### General Description

This course covers two days. The first day concentrates on programming, while the second concentrates on databases.

Unless you are a programmer by trade, programming may be a confusing discipline with terminology that is baffling, at best. This course is designed to introduce the tester and quality assurance professional to programming concepts. We'll discuss how programming relates to QA and what we need to know to be effective testers. Terminology is probably the most difficult hurdle in understanding the programming world, so we'll cover terms and their uses and provide you with enough information to be able to participate in technical discussions effectively. As each programming concept is investigated, particular note will be made of how this information affects us as testers. You won't leave this course being a programmer, but you will be a better-informed tester who can talk the talk and doesn't have to feel confused in a technical review.

Knowledge is power and the data is what creates the knowledge. We need to know how to store it and how to retrieve it in order to be effective in testing the data flow through a system. In this section we will discuss what a database is, why it's used and how to use it. A significant amount of time will be spent discussing SQL and walking through examples. SQL can be a major tool for verifying the data that you are processing during testing. Throughout this course we'll look at database design issues and how they affect the testing requirements.

### Learning Objectives

A successful instructor/attendee/material interaction will allow the student to:

- Understand the concepts behind programming languages
- Differentiate between programming languages and their most efficient uses
- Understand how data is handled and manipulated within programs
- Learn how programming has changed in the internet world
- Understand how programming relates to testing and why you need to know the basics

## Course Outline

- Understand what a database is and why it's used
- Become familiar with the most popular database tools
- Learn what data types are, why they are used and how they affect your test cases
- Grasp the concepts of database design
- Design test cases to effectively test for data handling, performance and recoverability
- Become an active participant in database design reviews

## Course Materials

This course includes the following materials:

<i>Name</i>	<i>Description</i>
Course Outline	A general description of the course along with learning objectives, course materials and an outline of the course topics, including approximate timings for each section.
Noteset	A set of approximately 400 PowerPoint slides covering the topics to be addressed.
Bibliography and resources	A set of further readings, Web sites, tools and other resources to help implement the concepts.

The printed course materials are provided in a binder in a way which makes it convenient for course attendees to remove portions as needed for reference; e.g., during exercises.

## Session Plan

### *Day One* - Programming

#### Introduction

- What are we going to cover
- Terminology

#### What is programming?

- What is a program
- Getting the instructions

#### Programming languages

## Course Outline

- Why so many?
- Machine language
- Assembly language
- High level language
- C
- C++
- Object Orientation
- Rapid application development (RAD)
- Database languages
- Scripting languages
- Web languages
- So what is the best language?

### Simple programming concepts

- Design is critical
- The programmer's toolkit
- Input/output
- Control statements
- Boolean operators
- Case statements
- Loops

### More programming concepts

- Structured programming
- Subprograms and modules
- Sorting and searching algorithms

### It's all about the data

- Long term storage
- Temporary complex data structures

### Internet programming particulars

- HTML
- Java
- How extreme is extreme programming?

### Programming as it relates to quality assurance

- Debugging vs. troubleshooting
- Logic behind the behavior
- Why is error handling hard?
- Programming vs. QA/tester skills
- What if the discussion gets too technical?
- QA involvement during design
- Target your testing

## Course Outline

### *Day Two* - Databases

#### What is a database?

- Flat files
- Hierarchical databases
- Relational databases

#### Why use one?

- Data integrity
- Data accessibility
- Predicting the future

#### Database tools

- RAD DBMS tools
- SQL

#### Identifying the data

- Data types and how to define them
- Numbers, strings and others

#### Organizing the data

- Who needs what, when
- Schemas and tables
- Rows and columns

#### Creating the database

- Database design
- Creating tables
- Example walkthroughs with RAD and SQL

#### Where's the data?

- Populating the database
- Accessing the data - Finding what you want
- Protecting the data

#### SQL in the modern world

- ODBC and JDBC
- Triggers and cursors
- Stored modules
- Handling errors

#### Database technology and how it relates to QA

- The data is what it's all about
- Integrity of data handling
- Performance and recoverability testing issues

## Course Outline

- Error handling testing issues
- QA involvement during design

## Recommended Readings

The class materials include an extensive bibliography of books related to software testing, project management, quality, and other topics of interest to the test professional.