Quality Assurance Training Program

Introduction/Summary: This 5-day course focuses on understanding and developing various skills required by QA Developer, preparing to use various tools and techniques to check and make sure that the defects are identified and corrected before software is released to users. This is a comprehensive and intensive course with plenty of illustrated examples and augmented with practical hands-on exercises.

Training Type: Online training/Classroom training

Audience: This course is intended for anyone with Masters in IT related fields with some Computer Science background.

Duration: 40 hrs

Training Timings: At this point all courses will be Weekend or Weekdays depending upon the availability of the Trainer. We are going to schedule some orientation sessions (depending upon how many people want to attend) and first class is always free.

Faculty:

- Landmark has a pool of highly experienced working professionals who serves as faculty. They have years of real industry experience.
- Candidates are always welcome to speak to faculty.
- We have 4 Full Time Trainers with our Company who are available for consultation.
- We have a Training Manager who will guide you in every way and our Training Manager will discuss with you on the best possible Options based on current market conditions.

Approach: Instructor-led participative lecture with group exercises.

Cost: This training programme is absolutely **free.**

Accomadation: If needed, Accomadation will be provided to the trainees. Advance refundable deposit of $1000.00 required.

During And After Training Period Landmark Technologies:

- During the training period we help updating your profile.
- After training you will be placed with our client companies which include Fortune 1000 companies.
- Once you are placed on project you will be paid competitive salary. Percentage option offered after 1 year.
- Landmark Technologies, Inc. also will cover cost of your insurance package on salary option.
- Salary review after every 6 months, Paid Leaves, Authorized Overtime, Performance Pay, Green Card sponsorship depending on your performance.

Referral Fee: Referral fee of $500/trainee, paid on placement.

421 7th Avenue, Suite 1007 New York, NY 1001 Phone: (212) 863 9741 Fax: (212) 504 8320
info@landmark-technologies.com

Objectives:

Software Quality Assurance involves the entire software development process - monitoring and improving the process, making sure that any agreed-upon standards and procedures are followed, and ensuring that problems are found and dealt with. It is oriented to 'prevention'. The course teaches the techniques necessary to develop and maintain a systematic, integrated software testing approach for your organization. Experience the science and the art of both functional and structural testing methods in an informal workshop setting. Keep your testing efforts on track while reacting to changing priorities, technologies, and user needs. Learn to think like an attacker so that you can add test cases to cover non-functional—often implied or missing—security requirements.

- How to bridge the gap of testing from a developer's perspective vs. from a tester's perspective
- An introduction to unit testing methods—What developers do
- Knowledge of Manual and Automated Testing
- Business jargon, definitions
- Unit, system, integration, and user acceptance development and testing.
- Bugs that unit tests can expose and ones that they will miss
- How to leverage (as a tester, not a developer) unit tests to find more bugs earlier
- How to plan testing on Agile Development projects
- Working knowledge of the Testing Tools
- Developing and Designing Test Cases
- New and sometimes forgotten strategies and test design methods including attack-based testing, model-based testing, and keyword-driven testing
- More effective test-case design with a black-box testing approach
- How to select the most effective practices to find bugs, optimize test planning and execution time
- Understanding of BRD(Business Requirement Document) and SRS (Software Requirement Specification)
- Type of Testing Models.
- Repeating test cases
- Extending functionality

Course Content:

Outline Module 1: Overview and Introduction

Quality Assurance Standards and Methodologies
Software Development Life Cycle / Rational Unified Process / SDLC
Basic introduction of QA vs. QC, SDLC, CMM, ISO, SIX SIGMA, testing terms
Manual testing: risk analysis, usage cases, test plan using real life project,
Test Life Cycle
Test Planning
Planning fundamentals

Outline Module 2: Overview of Software development process

Microsoft SDLC (System Development Life Cycle)

V-Model, Water Fall, Spiral Model, Spiral/Iterative/Incremental, Agile, eXtreme
Rapid Application Development (RAD)
Rational Unified Process
Extreme Programming
RUP (Rational Unified Process)
Object Oriented Concepts
Structured Query Language (SQL) / UNIX / LINUX/ XML
Development Phases: Unit, Integration, System and User
    Attributes of each type of testing
    When to execute each type of test

Outline Module 3: Intro to QA tools

Test Management Tools - Test Director / Quality Center (Business Process Testing)
Integration of Mercury Tools (WinRunner / Quick Test Professional/ Load Runner) with Test Director and Quality Center

Outline Module 4: Test Management Tool QC (Quality Center)

Complete working of QC filing the requirements
Test environments—issues and concerns
Analyzing requirements to develop test items
Reporting and managing defects
Tracking Defects
Writing test cases and Designing test cases and test procedures
Executing test cases and Reporting
Defect life cycle from Logging a defect to fixing a defect

Outline Module 5: Unit Testing & Testing Strategies

Common strategies for unit testing
Common unit testing frameworks
Regression testing methods and issues
Interoperability/Integration Testing
    Case study of a web-based application
API Testing
    Differences between API testing and Unit Testing
    Reasons and methods for API testing
    API test case design techniques

Outline Module 6: Automation Tool: WinRunner, LoadRunner and QTP

WinRunner
    Recording user actions
    Setting the initial condition
    Synchronizing scripts with the application
    Record modes and run modes
    Verifying the application
    Data driving and automated test
    Working with GUI MAP files
    TSL Syntax and Usage
        Functions, variables, constants and operators

- Adding user interaction to a script
- Flow control
- Sending messages to the Test Results window

LoadRunner

- Differences between loads, stress, performance testing
- Tools available in the market for load testing
- Planning effective load test
- Recording
- Enhancing scripts with run settings, recording options
- Transactions points
- Rendezvous points
- Using LR functions
- Using error logging options
- Correlation Parameters
- Check points
- How to play back script
- How to implement multiple actions
- Controller
  - What is scenario?
  - How to assign scripts to scenario?
  - Manual scenario creation
  - Defining and connecting load generators
  - Setting scenario specific run time settings
  - Selecting and configuring the performance monitors
  - Loading and over loading
  - Using RAMPUP/RAMP DOWN

QTP

- Introduction to QTP
- Recording, running and analyzing scripts
- Designing the scripts
- Debugging the scripts
- Creating checkpoints
- Standard checkpoints
- Text checkpoints
- Text area checkpoints
- Bitmap checkpoints
- Object Spy
- Object repository
- Parameterize test scripts
- Reusable actions

TestDirector

- Requirements
- Coverage analysis/ test coverage
- Characteristics of useful test case
- Creating reports
- Creating test sets

Running tests sets
Setting test set properties
Manual and automated test execution
Adding and running test hosts
Analyzing test result
Defect tracking
Finding effective defects

Recovery System / Data Driven Testing/ Web Testing
Use WinRunner Recovery Manager to create compound recovery scenarios

Outline Module 7: Database Testing / Load Testing/ Stress Testing

Checking the Database
Verifying database content, columns, and rows
Performing runtime record checks

Creating Workflows
Working with Files
Custom Objects
Learning the properties of custom objects
Mapping custom objects to standard classes
Handling a custom object that cannot be mapped

Functions
Creating functions and compiled modules (libraries)
Return codes and error handling
Adding functions to the Function Generator and startup script

Outline Module 8: Smarter Testing

Tips and techniques for improving testing
White-box vs. Gray-box vs. Black-box testing
Developer testing vs. Tester testing
Model-based/State-transition testing
Testing use cases
Acceptance, system, build/integration, and unit level test plans

Contact Us:

Landmark Technologies, Inc.
421 7th Avenue, Suite 1007
New York, NY 1001
Phone: (212) 863 9741
Fax: (212) 504 8320
training@landmark-technologies.com