

Performance Test Engineering

Software Tester Market Scope:

Software QA engineering jobs are on a healthy growth track, with an estimated 100,000 new positions set to be created through 2022. That's higher growth than is anticipated for several other skilled technology jobs, according to the Bureau of Labor Statistics.

More and more, QA engineers are a necessary part of any company's tech team. These positions are an integral part to any software development platform. The average national salary for a QA engineer is \$75,444, according to the Dice Salary Survey.

Course Overview:

This Performance Test Engineering with Database and Network Administration course is an intensive hands-on course, designed to introduce the complexities of software performance testing and delivers testing skills that participants can immediately apply back on the job. Using a real-world case study, students will encounter issues, decisions, and testing experiences comparable to those in real-world work environment. Working through a series of discussion-based exercises students will develop a workable strategy for performance testing an application/system. The focus of the exercises is on analysis of a situation and understanding the planning and design issues associated with performance testing. This course will also focus on problem analysis, tuning, debugging, or tools. This course extends beyond the simple use of a tool to look at the nature of performance testing, the challenges surrounding performance testing and how to overcome these challenges to make the performance testing more effective. This course focuses on what performance testing really is, how it should be carried out and how it should be managed.

This course will train Performance Testing Tools like HP Load Runner, HP Performance Center, JMeter, MS Visual Studio and monitoring tools like HP SiteScope, MS SCOM.

Hands-on exercises are used to demonstrate each feature and the student will gain first-hand experience in the lab required to become a successful performance test engineer

Prerequisites:

Students who completed at least one computer training program or have some work experience in IT field with some knowledge in computer Networking or Programming.

Course Duration (20 Weeks)

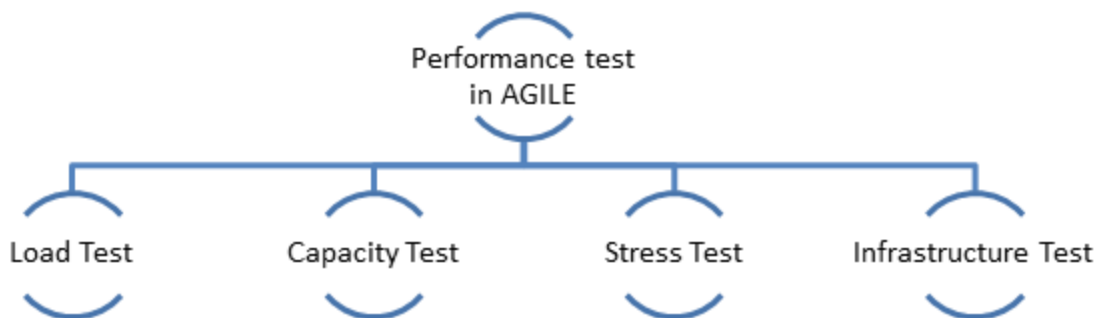
Class: 120 Hours | Instructor-Led Lab: 80 Hours | Total Hours: 200 Hours

Training Methodology:

- Real world scenario labs.
- Each student will be provided one Windows Server accessible from anywhere 24/7
- Each class is designed with Class Notes and Labs
- Resume Preparation
- Interview Training
- Job Support

Course Fee:

Tuition fee	\$2500
Payment plan	3 installments (\$1000 +\$1000+\$500)
1st installment	\$1000



Course Curriculum

Phase	Module	Description
Phase 1 Foundation Training	Module 1	Overview of enterprise applications and N-Tier Infrastructure
	Module 2	Operating System –Windows Server 2012/2016, Unix
	Module 3	Networking , Active Directory and DNS
	Module 4	Power Shell Scripting, Batch Scripting
	Module 5	Incident, Problem and Change Management process
Phase 2 Manual Testing	Module 6	Manual Testing <ul style="list-style-type: none"> Types of Testing, Agile Methodology and Live Project Overview SDLC and Test Scenarios Test Plan Complete Walkthrough Test case writing process, Test Data Test Execution, Traceability matrix, reviews, Status reporting Manual Testing on real world projects
	Module 7	Test Strategy and Defect Management <ul style="list-style-type: none"> Defect management Quality tools Change management version control
	Module 8	Test Management Tool <ul style="list-style-type: none"> JIRA Gits HP ALM
	Module 9	Backend Testing <ul style="list-style-type: none"> Database overview and SQL Database schema tests Stored Procedure tests Integration tests of procedures Trigger tests Integration tests of SQL server Checking data integrity and consistency Test back end via front end
	Module 10	Database tools <ul style="list-style-type: none"> TOAD Oracle SQL Developer
Phase 3 Automation Testing	Module 11	Performance Testing Tool – JMeter <ul style="list-style-type: none"> JMeter Architecture Loading Testing using JMeter Regular expressions and data driving

		<ul style="list-style-type: none"> • Handling dynamic responses • Bean shell scripting basics for JMeter • Integration of Selenium with JMeter • REST API Load Testing using JMeter • Database Testing using JMeter • Blaze meter plugin and JMeter Template • Monitoring Server Performance
	<p>Module 12</p>	<p>Performance Testing Tool – HP Load Runner</p> <p>Overview of Load Testing</p> <ul style="list-style-type: none"> • Introduction to web applications performance testing • Performance testing concepts : Stress, Load , Regression • What is Load Testing ,why real world need Load Testing • Overview of Tier-3 Enterprise Application infrastructure • Overview of Software Testing Environment- Dev , UAT, Pre-prod , Prod • Test Plan and Test Case <p>Overview of HP Load Runner</p> <ul style="list-style-type: none"> • Load Runner installation on a virtual server • Describe the Load Runner environment and architecture. • Explain how to install the Controller. • Explain how to install Load Generator • Explain how to add multiple LG to Controller <p>Load Runner Scripting</p> <p>Planning a Script</p> <ul style="list-style-type: none"> • Determining what steps to perform • Consider data and data usage Recording a Script • Recording Options – HTML vs. URL recording • Recording the test case <p>Transactions</p> <ul style="list-style-type: none"> • Add Transactions during recording • Add Transactions after recording using the transaction workflow tool, and manually via the keyword view. <p>Verifications</p> <ul style="list-style-type: none"> • Why is a verification point necessary • Add a verification during recording • Add a verification after recording using the keyword view screenshots • Global Verification Points <p>Run-Time Settings</p> <ul style="list-style-type: none"> • Setting appropriate logging option • Controlling the number of iterations to playback • Think times and Pacing settings

		<ul style="list-style-type: none"> • Other settings <p>Parameterization</p> <ul style="list-style-type: none"> • Adding Parameters in the keyword view, and in the script view • Create a file type parameter • Examine other parameter types, e.g. Random type, date time type • File parameter access methods • Update Parameter value settings • Simulate Parameter Usage <p>Correlation</p> <ul style="list-style-type: none"> • Required Settings • Run the test, examine errors • Use the Scan for Correlations tool • Determine which values to correlate • Review the correlation functions added to the script <p>Manual Correlations</p> <ul style="list-style-type: none"> • Run the test & examine the errors to determine which commands to focus on • Determine which of the values is the dynamic one • Determine where the server sent the dynamic value • Determine appropriate left and right boundaries and ordinal values • Add the web_reg_save_param function • Replace hard-coded dynamic values with the new parameter • Run the test to validate Correlation during Recording <p>Add a rule to the recording options</p> <ul style="list-style-type: none"> • Test the rule • Record the script • Use of the regenerate script option <p>Advanced Scripting</p> <ul style="list-style-type: none"> • Condition statements and loops • Variable declarations • Common string manipulation functions • Data conversion functions • LoadRunner specific functions (lr_eval_string, output messages, lr_save_string) Custom Checkpoints • Examining the web_reg_find savecount option • Capturing data using the web_reg_save_param function
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

		<p>String Manipulation</p> <ul style="list-style-type: none"> • Using the sprintf function • strstr • strcpy • strcat • lr_save_string and lr_save_int • lr_save_datetime function • lr_advance_param vs. lr_next_row <p>Scenario configuration</p> <ul style="list-style-type: none"> • Configure different types of scenarios. • Define a load test schedule using the Scheduler. • Set up a Service Level Agreement. • Set up Performance Monitors. • Configure Scenario options based on test requirements. • Configure additional features • Identify steps within the Scenario Execution Process. • Work with the Controller Run tab. • Describe results collation. <p>Results analysis</p> <ul style="list-style-type: none"> • Use Root Cause Analysis to determine the likely cause of a bottleneck. • Analyze information on available graphs. • Use Drill-Down to determine lower level details. • Use Auto Correlation to identify probable cause. • Use filtering, set granularity to analyze results. • Compare or merge graphs to analyze results. • Describe the uses of the SLA and Analyze Transaction feature. • Describe Analysis module features under Options. • Describe how to use cross-scenario analysis to evaluate impacts of application • Explain the reporting solution. • Identify diagnostic graphs.
	<p>Module 13</p>	<p>Load Testing with MS Visual Studio</p> <p>Overview of the web performance and Load test</p> <ul style="list-style-type: none"> • How to configure your visual studio environment to start web performance and Load test • Run results and how to analyzing the load performance result <p>Scenario Information</p> <ul style="list-style-type: none"> • Load pattern (constant load, step load) • Test Mix Model based on (total no. of tests, no. of virtual users, user pace or sequential test order) • Test Mix (add no. of previously created web tests and determined the distribution %)

		<ul style="list-style-type: none"> • Network Mix (add multiple network types and determined the distribution %) • Browser Mix (add multiple browser types and determined the distribution %) <p>Prerequisite steps to start your Load test</p> <ul style="list-style-type: none"> • Start to record many web performance tests that reflect your test cases that cover the user business cases • Each web test should cover only one test case (Flow) • Check and validate each flow separately
	Module 14	<p>Monitoring Tools</p> <p>Overview of Performance Testing Monitoring</p> <ul style="list-style-type: none"> • HP Site Scope • MS SCOM • JConsole • CA Wily • Monitoring Java web application • Collecting and analyzing Java dump file to find out the memory leak <p>Web Server</p> <ul style="list-style-type: none"> • IBM Web Sphere Application Server – Installation and Configuration • Tomcat - Installation and Configuration • IIS - Installation and Configuration • Java Application deployment on Web Sphere- Installation and Configuration
	Module 15	<p>API Testing</p> <ul style="list-style-type: none"> • Functionality testing • Reliability testing • Load testing • Creativity testing • Security testing and access controls. • Proficiency testing
	Module 16	<p>Security Testing</p> <ul style="list-style-type: none"> • Fundamentals of Security Testing • Web application coding security • Web application security testing • Ethical hacking of applications • Web portal security testing
Phase 4 Real World Project	Module 17	<p>Boot Camp</p> <ul style="list-style-type: none"> • Boot Camp with real world project

<p>Phase 5 Job Placement</p>	<p>Module 18</p>	<p>Job Placement</p> <ul style="list-style-type: none">• Resume Writing• Project Analysis• Interview Preparation• Mock Interview
-----------------------------------------	------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

To download the complete course catalog, please visit:

<http://training.digitalpoint.tech>

Email: admin@digitalpoint.tech