* Question 1. What Is Jmeter?

Answer :

JMeter is one of the Java tools which is used to perform load testing client/server applications. Apache JMeter is open source software, a 100% pure Java desktop application designed to load test functional behavior and measure performance of the application. It was originally designed for testing Web Applications but has since expanded to other test functions.

* Question 2. Explain How Jmeter Works?

Answer :

JMeter acts like a group of users sending requests to a target server. It collects response from target server and other statistics which show the performance of the application or server via graphs or tables.

* Question 3. Explain Where You Can Use Functions And Variables?

Answer :

Variables and functions can be written into any field of any test component.

* Question 4. Mention What Are Regular Expressions In Jmeter?

Answer :

Based on the pattern, regular expression are used to search and manipulate text. JMeter is useful in interpreting forms of regular expression or patterns being used throughout a JMeter test plan.

* Question 5. Explain What Is Samplers And Thread Groups?

Answer :

Thread group: For any test plan, JMeter is the beginning part of thread group elements. It is an important element of JMeter, where you can set number of users and time to load all the users given in the thread group

Samplers: Sampler generates one or more sample results; these sample results have many attributes like elapsed time, data size, etc. Samplers allow JMeter to send specific types of requests to the server, through samplers, thread group decides which type of request it need to make. Some of the useful samplers are HTTP request, FTP request, JDBC request and so on.

* Question 6. Whether The Test Plans Built Using Jmeter Are Os Dependent?

Answer :

Usually, test plan are saved in their XML format, so there is nothing to do with any particular O.S. It can be run on any OS where JMeter can run.

* Question 7. Mention What Are The Types Of Processor In Jmeter?

Answer :

The types of processor in JMeter are:

1. Pre-processor
2. Post processor

* Question 8. Explain What Are Pre-processor Elements? List Some Of The Pre-processor Elements?

Answer :

A pre-processor is something that will happen before sampler executes. To configure the sample request prior to its execution or to update variables that are not extracted from response text pre-processor elements are used.

Some of the pre-processor elements are:

* HTTP URL re-writing modifier
* HTTP user parameter modifier
* HTML link parser
* BeanShell PreProcessor
* Question 9. Mention The Execution Order Of Test Elements?

Answer :

The test plans elements execution order is

* Configuration elements
* Pre-processors
* Timers
* Samplers
* Post-processors
* Assertions
* Listeners
* Question 10. What Does “contain” And “matches” Indicates In The Regular Expression?

Answer :

In the regular expression, contains indicates that the regular expression matched at least some part of the target. While matches means the regular expression matched the whole target. So, ‘alphabet’ is “matched” by ‘al.\*t.’

* Question 11. Explain What Is Configuration Elements?

Answer :

A configuration element works parallel with a Sampler. To set up defaults and variables for later use by samplers configuration elements can be used. At the start of the scope, these elements are processed before any samplers in the same scope.

* Question 12. Explain What Is A Timer In Jmeter And What Are The Types Of It?

Answer :

A JMeter thread by default will send requests continuously without any pause. To get a pause between the request, Timers are used. Some of the Timers used are Constant Timer, Gaussian Random Timer, Synchronizing Timer, Uniform Random Timer and so on.

* Question 13. Explain What Is Test Fragment?

Answer :

Test fragment is also a type of element like Thread Group element. The only difference is test fragment is not implemented unless it is referenced by either a Module controller or an Include controller.

* Question 14. Explain What Is Assertion In Jmeter? What Are The Types Of Assertion?

Answer :

Assertion helps to verify that your server under test returns the expected results

Some commonly used Assertion in JMeter are

* Response Assertion
* Duration Assertion
* Size Assertion
* XML Assertion
* HTML Assertion
* Question 15. Explain How You Can Reduce The Resource Requirement In Jmeter?

Answer :

To reduce the resource requirements in JMeter

* Use non-GUI mode: jmeter –n –t test.jmx –l test.jtl
* During the load, test doesn’t use “view results tree” or “view results in table” listeners, use them only during scripting phase
* Don’t use functional mode
* Instead of using lots of similar samplers, use the same sampler in loop and use variable to vary the sample
* Question 16. Explain How You Can Perform Spike Testing In Jmeter?

Answer :

By synchronizing, timer JMeter spike testing can be achieved. Synchronizing timer, blocks thread until a specific amount of threads has been blocked and then release them all together thus creating large instantaneous load.

* Question 17. Explain How You Can Capture The Script Of The Authentication Window In Jmeter?

Answer :

Normally, you can capture script by recording.

* First you have to Threadgroup in Testplan and then make HTTPProxyServer in Workbench
* After that, set port number in Global Setting box (e.g., 8911) and modify your connection setting in IE as local host in address 8911 as in port Then you can start http proxy server in JMeter and run your application for login.
* Question 18. List Out Few Jmeter Listeners?

Answer :

Some of the JMeter Listeners are:

* Spline Visualizer
* Aggregate Report
* View Result Tree
* View Result in Table
* Monitor Results
* Distribution Graph
* BeanShell Listener
* Summary Report and so on
* Question 19. What Is Distributed Load Testing? How It Can Be Achieved?

Answer :

Distributed load testing is the process through which numerous systems can be used for simulating load of a large number of users. By using the master-slave configuration, JMeter can do distribute load testing.

* Question 20. In Jmeter Is It Necessary To Call Embedded Resources Explicitly?

Answer :

You can eliminate all embedded resources from being explicitly called. Requests have a checkbox at the bottom that says “retrieve embedded resources.” It would grab all CSS, JPG, etc. It is a brilliant way to find resources and broken link in a web App.

* Question 21. Explain What Is The Role Of Timer In Jmeter?

Answer :

With the help of timer, JMeter can delay the time between each request, which a thread makes. It can solve the overload problem of the server.

* Question 22. Explain What Is Post-processor?

Answer :

To perform any action after making a request, Post- processor is used. For example, if JMeter sends an http request to the web server, and if you want JMeter to stop sending the request if the web server shows an error, then you will use post-processor to perform this action.

* Question 23. What Are The Benefits That Jmeter Offers For Performance Testing?

Answer :

JMeter offers benefits on performance testing like

* It can be used to test performance for both, static resources as well as dynamic resources
* It can handle a maximum number of concurrent users then your website can handle
* It provides the graphical analyses of performance reports
* Question 24. What Are The Protocols Supported By Jmeter?

Answer :

The protocols supported by JMeter are:

* Web: HTTP, HTTPS sites 'web 1.0' web 2.0 (ajax, flex and flex-ws-amf)
* Web Services: SOAP / XML-RPC
* Database via JDBC drivers
* Directory: LDAP
* Messaging Oriented service via JMS
* Service: POP3, IMAP, SMTP
* FTP Service
* Question 25. List Some Of The Features Of Jmeter.?

Answer :

Following are some of the features of JMeter:

* Its free. Its an open source software.
* It has simple and intuitive GUI.
* JMeter can load and performance test many different server types: Web - HTTP, HTTPS, SOAP, Database via JDBC, LDAP, JMS, Mail - POP3
* It is platform-independent tool. On Linux/Unix, JMeter can be invoked by clicking on JMeter shell script. On Windows it can be invoked by starting the jmeter.bat file.
* It has full Swing and lightweight component support (precompiled JAR uses packages javax.swing.\* ).
* JMeter store its test plans in XML format. This means you can generate a test plan using a text editor.
* It's full multi-threading framework allows concurrent sampling by many threads and simultaneous sampling of different functions by separate thread groups.
* It is highly Extensible.
* Can also be used to perform automated and functional testing of your application.
* Question 26. What Is A Test Plan In Jmeter?

Answer :

A Test Plan defines and provides a layout of how and what to test. For example the web application as well as the client server application. It can be viewed as a container for running tests. A complete test plan will consist of one or more elements such as thread groups, logic controllers, sample-generating controllers, listeners, timers, assertions, and configuration elements. A test plan must have at least one thread group.

* Question 27. List Some Of The Test Plan Elements In Jmeter.?

Answer :

Following is a list of some of the test plan elements:

* ThreadGroup
* Controllers
* Listeners
* Timers
* Assertions
* Configuration Elements
* Pre-Processor Elements
* Post-Processor Elements
* Question 28. What Is Thread Group?

Answer :

Thread Group elements are the beginning points of your test plan. As the name suggests, the thread group elements control the number of threads JMeter will use during the test.

* Question 29. What Are Controllers And Its Types?

Answer :

JMeter has two types of Controllers:

Samplers Controllers : Samplers allow JMeter to send specific types of requests to a server. They simulate a user's request for a page from the target server. For example, you can add a HTTP Request sampler if you need to perform a POST, GET, DELETE on a HTTP service

Logical Controllers : Logic Controllers let you control order of processing of Samplers in a Thread. Logic Controllers can change the order of request coming from any of their child elements. Some examples are: ForEach Controller, While Controller, Loop Controller, IF Controller, Run Time Controller, Interleave Controller, Throughput Controller, Run Once Controller.

* Question 30. What Is Configuration Element?

Answer :

Configuration Elements allow you to create defaults and variables to be used by Samplers. They are used to add or modify requests made by Samplers.

They are executed at the start of the scope of which they are part, before any Samplers that are located in the same scope. Therefore, a Configuration Element is accessed only from inside the branch where it is placed.

* Question 31. What Are Listeners?

Answer :

Listeners let you view the results of Samplers in the form of tables, graphs, trees or simple text in some log files. They provide visual access to the data gathered by JMeter about the test cases as a Sampler component of JMeter is executed.

Listeners can be added anywhere in the test, including directly under the test plan. They will collect data only from elements at or below their level.

* Question 32. What Are Pre-processor And Post-processor Elements?

Answer :

A Pre-Procesor is something that will happen before a sampler executes. They are often used to modify the settings of a Sample Request just before it runs, or to update variables that are not extracted from response text.

A Post Processor executes after a sampler finishes its execution. This element is most often used to process the response data, for example, to retrieve particular value for later use.

* Question 33. What Is The Execution Order Of Test Elements?

Answer :

Following is the execution order of the test plan elements:

* Configuration elements
* Pre-Processors
* Timers
* Sampler
* Post-Processors (unless SampleResult is null)
* Assertions (unless SampleResult is null)
* Listeners (unless SampleResult is null)
* Question 34. How Do You Ensure Re-usability In Your Jmeter Scripts?

Answer :

* Using config elements like "CSV Data Set Config", "User Defined Variables", etc for greater data reuse.
* Modularizing shared tasks and invoking them via a "Module Controller".
* Writing your own BeanShell functions, and reusing them.
* Question 35. Are The Test Plans Built Using Jmeter Os Dependant?

Answer :

Test plans are usually saved in thr XML format, hence they have nothing to do with any particular OS. You can run those test plans on any OS where JMeter can run.

* Question 36. What Are The Monitor Tests?

Answer :

Uses of monitor tests are:

* Monitors are useful for a stress testing and system management.
* Used with stress testing, the monitor provides additional information about server performance.
* Monitors makes it easier to see the relationship between server performance and response time on the client side.
* As a system administration tool, the monitor provides an easy way to monitor multiple servers from one console.
* Question 37. What Are Jmeter Functions?

Answer :

JMeter functions are special values that can populate fields of any Sampler or other element in a test tree.   
A function call looks like this:

${\_\_functionName(var1,var2,var3)}

* Question 38. Where Can Functions And Variables Be Used?

Answer :

Functions and variables can be written into any field of any test component.

* Question 39. What Are Regular Expressions In Jmeter?

Answer :

Regular expressions are used to search and manipulate text, based on patterns. JMeter interprets forms of regular expressions or patterns being used throughout a JMeter test plan, by including the pattern matching software Apache Jakarta ORO.

* Question 40. How Can You Reduce Resource Requirements In Jmeter?

Answer :

Below are some suggestion to reduce resource requirements:

* Use non-GUI mode: jmeter -n -t test.jmx -l test.jtl.
* Use as few Listeners as possible; if using the -l flag as above they can all be deleted or disabled.
* Disable the “View Result Tree” listener as it consumes a lot of memory and can result in the console freezing or JMeter running out of memory. It is, however, safe to use the “View Result Tree” listener with only “Errors” checked.
* Rather than using lots of similar samplers, use the same sampler in a loop, and use variables (CSV Data Set) to vary the sample. Or perhaps use the Access Log Sampler.
* Don't use functional mode.
* Use CSV output rather than XML.
* Only save the data that you need.
* Use as few Assertions as possible.
* Disable all JMeter graphs as they consume a lot of memory. You can view all of the real time graphs using the JTLs tab in your web interface.
* Do not forget to erase the local path from CSV Data Set Config if used.
* Clean the Files tab prior to every test run.