1. **What is Performance Testing?**

Performance testing, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as Speed, scalability and Stability.

* 1. Speed - Determines whether the application responds quickly
  2. Scalability - Determines maximum user load the software application can handle.
  3. Stability - Determines if the application is stable under varying loads

1. **Types of Performance Testing?**
   1. Load Testing - Load Testing is type of performance testing to check system with constantly increasing the load on the system until the time load is reaches to its threshold value means the time at which application sends slow responses and speed of the application decreases.
   2. Stress Testing - Stress Testing is performance testing type to check the stability of software when hardware resources are not sufficient like CPU, memory, disk space etc. means the time at which application gets crash due to un-handling capability of users.
   3. Volume Testing - Volume testing is non-functional testing which refers to testing a software application with a large amount of data to be processed to check the efficiency of the application. The main goal of this testing is to monitor the performance of application under varying database volumes.
   4. Scalability Testing - The objective of scalability testing is to determine the software application's effectiveness in "scaling up" to support an increase in user load. It helps plan capacity addition to your software system.
   5. Spike testing - tests the software's reaction to sudden large spikes in the load generated by users.
   6. Endurance testing - is done to make sure the software can handle the expected load over a long period of time.
2. **What is Jmeter and Advantages of Jmeter over other performance testing tools?**

JMeter is an open source application. JMeter is an software that can be used to execute performance testing, load testing and functional testing of your web applications. JMeter can also simulate a heavy load on a server by creating tons of virtual concurrent users to web server.

* 1. Open Source: JMeter is an open source software. This means that it can be downloaded free of cost. It is also a 100% pure Java application.
  2. Ease of Use: The user can install and use JMeter easily. Just download it from internet, install and run. As a pure Java desktop application, it comes ready to use with default settings. It does not require you to have any specific skills or domain knowledge to use it.
  3. Robust Reporting: JMeter can generate the effective reporting. The test result can be visualized by using Graph, Chart, and Tree View. JMeter supports different formats for reporting like text, XML, HTML and JSON
  4. Ultimate Testing: With JMeter, user can do any kind of testing they want. Load Test, Stress Test, Functional Test, Distributed Test, all in one tool.
  5. Flexibility: You can customize JMeter as per your requirement and apply the automation testing to JMeter. You can save the effort of executing test cases manually.

1. **What is Thread Group in Jmeter?**

Thread group elements are the beginning points of any test plan. A thread group is a set of threads executing the same scenario. Set the number of iterations in the configuration. Thread behaviour is defined according to ramp up and destroyed once the number of iterations per thread has elapsed. The controls for a thread group allow you to:

* 1. Set the number of threads
  2. Set the ramp-up period
  3. Set the number of times to execute the test

1. **What is Thread Group in Jmeter?**

[Listeners](http://jmeter.apache.org/usermanual/listeners.html) are the [JMeter](https://www.blazemeter.com/jmeter-load-testing?utm_source=BM&utm_medium=BM_blog&utm_campaign=jmeter-listeners-part-basic-display-formats) component that displays test results.

1. View Results Tree: The View Results Tree listener displays samples that the JMeter samplers generate, and the assertion results that are related to these samples. This listener displays the samples in the order they are generated by the JMeter script ,and provides parameters and data for each of them.
2. Response Graph Time: An additional listener that plots a graph of the samples’ response time over the course of the test is the Response Time Graph listener. This listener is able to plot a graph either for all samples or for selected ones.
3. Assertion Results: TheAssertion Results listener displays results of all the assertions in its scope. This listener displays samples as they go and failed assertions for related samples, if there are any. Passed assertions are not shown.
4. Generate Summary Result: This test element can be placed anywhere in the test plan. Generates a summary of the test run so far to the log file and/or standard output. Both running and differential totals are shown.
5. View Result Table: This visualizer creates a row for every sample result. Like the View Results Tree, this visualizer uses a lot of memory.
6. **Explain How do you recorded/made your script in your language?** After opening jmeter gui, gui will automatically creates test plan. **Step 1**: we need to right click on test plan, select add, select Threads (users), click on Thread Group (Name can be given by user ex.Login). **Step 2:** we need to right click on Thread group, select add, select Logic Controller, and click on Recording controller. **Step 3:** right click on Thread Group, select add, select Listener, and click on View Results Tree. **Step 4:** click on Test Plan, select add, select Non-Test Elements, and click on HTTP(s) Test Script Recorder. **Step 5:** After adding HTTP(s) Test Script Recorder, In that Recorder we need request filter so that unnecessary files will not include in script. **Step 6:** We need to set port number that will be used while hitting url into browser. Then we need to add certificates into browser. We also need to add proxy setting having port 8080 and server localhost. **Step 7:** After all these steps: In HTTP(s) Test Script recorder we need to click on start. Then into the browser we need to pass url which we want make script of that particular operation. **Step 8:** After passing values and steps. We need to stop Recorder. In Recording Controller, you’ll see script is generated.
7. **Why do we need to do performance testing?**

Performance Testing is done to provide stakeholders with information about their application regarding speed, stability and scalability. More importantly, Performance Testing uncovers what needs to be improved before the product goes to market. Without Performance Testing, software is likely to suffer from issues such as: running slow while several users use it simultaneously, inconsistencies across different operating systems and poor usability. Performance testing will determine whether or not their software meets speed, scalability and stability requirements under expected workloads. Applications sent to market with poor performance metrics due to non existent or poor performance testing are likely to gain a bad reputation and fail to meet expected sales goals. Also, mission critical applications like space launch programs or life saving medical equipments should be performance tested to ensure that they run for a long period of time without deviations.