1. What is Performance Testing?

**Performance Testing** is performed to evaluate the performance of components of a particular system under a particular workload. During this testing, system components are monitored to verify the stability of the system under test. Performance Testing is a type of testing to ensure software applications will perform well under their expected workload. Features and Functionality supported by a software system is not the only concern. A software application's performance like its response time, reliability, resource usage and scalability do matter. The goal of Performance Testing is not to find bugs but to eliminate performance bottlenecks.

The focus of Performance Testing is checking a software program's

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

1. Types of Performance Testing?

* **Load Testing:**  is a type of performance testing conducted to evaluate the behavior of a system at increasing workload. It is the simplest form of testing conducted to understand the behavior of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, application server, etc., are also monitored.
* **Stress Testing:** It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.
* **Soak testing -**Soak Testing also known as endurance testing, is performed to determine the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues. The main aim is to discover the system's performance under sustained use.
* **Spike testing -**Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the workload.

1. What is Jmeter and Advantages of Jmeter over other performance testing tools?

* JMeter is a 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.
* JMeter is an open source application. This means that you can download JMeter’s source code to analyze and modify it if you want.
* Stefano Mazzocchi of the Apache Software Foundation first designed it. He is now a software engineer at Google. Nowadays, JMeter has become one of the most popular testing tools in the world, beside Selenium and Load Runner.

when we are doing testing for a web application, we need to know that what would be the performance of an application when numbers of user's use it concurrently. for this purpose, we use performance testing tolls like JMeter, Load Runner, Gatling or many other.

* **Benefits Of JMeter:-**

1. JMeter Graphics is easy to use  
2. Attributes well indicated in JMeter  
3. Easy to add a test scenario.  
4. No scripting required for it.  
5. JMeter is platform-independent.  
6. It has an open source code.   
7. Licence not required, It's a free to use.  
8. It's an open source application, Anyone can inspect, modify and enhance its source code.  
9. JMeter has conceptualized Test Results.  
10. JMeter supports plugins, so that you can write your own tests.  
11. It supports multiple protocols.

1. What is Thread Group in Jmeter?

A Thread Group in JMeter represents a pool of virtual users performing a set of operations. For example, considering Google search, one set of users will be using the search by text functionality, others will be using News Search, some other smaller section of users might be using search by image functionality. While creating a performance test script, for all these users, we will create different Thread Groups in JMeter with different thread count based on the user distribution across the functionalities. The different types of sampler like HTTP requests are added as child of these Thread Group elements, simulating the requests made by users to the server.

1. Write down and explain any 5 listeners present in Jmeter?

Listeners are used to display the results of test execution so that testers get to know the stats. We have around 15 listeners in Jmeter but mostly used ones are table, tree, and Graph.

1. **View Results In Table:**

This is the most commonly used and easily understandable form of listeners. It displays the result in form of table with some important performance parameters.

Listeners can be added directly under Test plan or under a sampler. The difference is, when you add listener under a sampler, it will show the results of that sampler only.  If we add sampler directly under the test plan, it displays the result for all the Sampler up in the hierarchy.

1. **View Results in Tree:**

This is another most commonly used listeners and provides detailed information with request and response. One can also view the HTML page rendered in response apart from viewing Json, XML, Text, RegEx.It is very useful as testers can put assertions on the response received to be ensured that the test passed. Jmeter results still show “Pass” even if the response is not desired.

1. **Work Bench:**

A workbench is a place where you can store those elements which are not in use in your current test plan but which can be later copy pasted in it. When you save JMeter file, the components which are present in workbench are not automatically saved. You must save them separately by right clicking and choose “Save as “option.all might be thinking then what is the use of workbench, anyways it is easy to add any component directly to a Jmeter’s Test Plan.The reason of having workbench was that user could do some experiments and try out new scenarios.

**They are listed here:**

* HTTP Mirror Server
* HTTP(s) Test Script Recorder
* Property Display

1. **Assertion:**

To ensure that the response received is correcr and as per expectation, we need to add assertion. Assertion are simply validations which are need to put on responses to compare the results.

***Below are the types of assertions commonly used:***

* Response Assertion
* Duration Assertion
* Size Assertion
* XML Assertion
* HTML Assertion

1. **Response Assertion:**

In Response Assertion, we can add our own pattern strings and then compare them with the responses received from a server. For example, you all know response code is 200 when any request returns some response successfully. So, if we add pattern string “Response Code = 202” then the test case should fail.

1. Explain How do you recorded/made your script in your language?
   * 1. Start jmeter
     2. select Test plan →Add → Thread Group(login)
     3. click on logic → add→ logic controller → Recording controller
     4. Test Plan→ Add → Non-Test Elements → HTTP(s) Test Script Recorder
     5. login → add → Listener → view results tree
     6. HTTP(s) Test Script Recorder then click on start button
     7. Now Set the HTTP proxy and port number
     8. Add certificate
     9. Now Hit [accounts-dev.nfhs.org](http://accounts-dev.nfhs.org/) this website
     10. After the login and logout from this website click on stop button of HTTP(s) Test Script Recorder.
     11. Check the list of recording controller in jmeter
     12. add the Regular Expression Extractor in sing\_in recording set the variables and regular expression
     13. Now change the authorization in validate recording
     14. now in view result tree start the execution of recording and see the script whether it pass or failed
2. 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.