1- What is the difference between manual testing and automated testing ?

The biggest difference between manual and automation testing is who executes the test case. In manual testing, the human tester does it. In automation testing, the tool does it.

2- What does Assert class ?

Assert is a method useful in determining Pass or Fail status of a test case, The assert methods are provided by the class org.junit.Assert which extends java.lang.Object class. There are various types of assertions like Boolean, Null, Identical etc.

3- How can be tested 'private' methods ?

We don’t need to test those private methods individually.

To test private methods, we just need to test the public methods that call them. We can call our public method and make assertions about the result or the state of the object. If the tests pass, we know our private methods are working correctly.

4- What is Monolithic Architecture ?

Monolithic architecture is the traditional unified model for the design of a software program. Monolithic, in this context, means composed all in one piece.

5- What are the best practices to write a Unit Test Case ?

1)Adopt a well-organized test practice.

2)Name your test well.

3)Write reliable and trustworthy unit tests.

4)Make automated unit testing a rule.

5)Focus on single use-case at a time.

6)Minimal assertion per test.

7)Unit test should be isolated.

8)Truly unit, not integration.

9) Aim for % 100 code coverage

10) Start using headless testing in the cloud

6- Why does JUnit only report the first failure in a single test ?

Reporting multiple failures in a single test is generally a sign that the test does too much and it is too big a unit test. JUnit is designed to work best with a number of small tests. It executes each test within a separate instance of the test class.

7- What are the benefits and drawbacks of Microservices ?

Benefits :

Greater agility

Faster time to market

Better scalability

Faster development cycles (easier deployment and debugging)

Easier to create a CI/CD pipeline for single-responsibility services

Isolated services have better fault tolerance

Platform- and language agnostic services

Cloud-readiness

Drawbacks :

Needs more collaboration (each team has to cover the whole microservice lifecycle)

Harder to test and monitor because of the complexity of the architecture

Poorer performance, as microservices need to communicate (network latency, message processing, etc.)

Harder to maintain the network (has less fault tolerance, needs more load balancing, etc.)

Doesn’t work without the proper corporate culture (DevOps culture, automation practices, etc.)

Security issues (harder to maintain transaction safety, distributed communication goes wrong more likely, etc.)

8- What is the role of actuator in spring boot ?

Actuator is mainly used to expose operational information about the running application — health, metrics, info, dump, env, etc. It uses HTTP endpoints or JMX beans to enable us to interact with it. Once this dependency is on the classpath, several endpoints are available for us out of the box.

9- What are the challenges that one has to face while using Microservices ?

Challanges of microservicChallanges of microservices are :

Design.

Security.

Testing.

Increased operational complexity.

Communication.

Your defined domain is unclear/uncertain.

Improved efficiency isn't guaranteed.

Application size is small or uncomplex.

10- How independent microservices communicate with each other?

Because of microservices are distributed and microservices communicate with each other by inter-service communication on network level. Each microservice has its own instance and process. Therefore, services must interact using an inter-service communication protocols like HTTP, gRPC or message brokers AMQP protocol.

11- What do you mean by Domain driven design ?

Domain-Driven Design is an approach to software development that centers the development on programming a domain model that has a rich understanding of the processes and rules of a domain. The name comes from a 2003 book by Eric Evans that describes the approach through a catalog of patterns.

12- What is container in Microservices ?

Containers are a form of operating system virtualization. A single container might be used to run anything from a small microservice or software process to a larger application. Inside a container are all the necessary executables, binary code, libraries, and configuration files.

13- What are the main components of Microservices architecture ?

Main components of Microservices architecture are :

- Clients.
- Identity Providers.
- API Gateway.
- Messaging Formats.
- Databases.
- Static Content.
- Management.
- Service Discovery.

14- How does a Microservice architecture work?

Microservice architecture, aka microservices, are a specific method of designing software systems to structure a single application as a collection of loosely coupled services. Applications tend to begin as a monolithic architecture , and over time grow into a set of interconnected microservices.