HW#6

1) What is the difference between manual testing and automated testing?
Automated testing uses automated tool for testing. But at manual testing, all tests are executed by tester. Automated testing is more reliable and faster than manual testing. Manual testing allows to human observation but automated testing does not require it.

2) What does Assert class?
Assert class provides assert methods. Assert methods helps to provide writing test cases and detect test failure. Using exception for rare errors slow down to programme. In such cases it is better to use asset class to prevent errors.

3) How can be tested 'private' methods?
To test private methods, it is sufficient that we test the public methods that call them. If the public methods pass tests, we know our private methods are working correctly.

4) What is Monolithic Architecture?
Monolithic Architecture means self-contained software design. In a monolithic application, all component are built as single code base and deployed as a single file. Because of it is single codebase, components are less loose couple.

5) What are the best practices to write a Unit Test Case?
Unit tests should be trustworthy, maintainable, readable, isolated, automated. Unit tests should verify a single-use case. Unit tests should be executed within an organized test practice.

6) Why does JUnit only report the first failure in a single test?
It is about Junit design decision. When we decide to report multiple failures per test, we begin to fight against JUnit.

7) What are the benefits and drawbacks of Microservices?
Benefits of Microservices: Microservices are independent applications of each other and focused on a single job. So we can develop every services different languages. This provides that a application is not depend one language. Also using microservices saves time. Because build and deploy time of services is different each other.
Drawbacks of Microservices: Microservies communications provide with messaging. Communication can be hard without using automation and advanced methodologies such as Agile.

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.

9) What are the challenges that one has to face while using Microservices?
Designing microservices is complex. Microservices are often deployed across multi-cloud environments. This cause vulnerable points in security.

10) How independent microservices communicate with each other?
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.
Microservices Communication types: Synchronous and Asynchronous Communication

11) What do you mean by Domain driven design?
Domain Driven Design is an approach that helps us solve and manage the complexity in our project, and also allows us to make our project sustainable. The complexity is we can have a lot of business rules in our project.

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?
Clients, identity providers, API gateway, messaging formats, databases, static content, management, service discovery.

14) How does a Microservice architecture work?
Microservices architecture focuses on classifying large, bulky applications. Each microservice is designed to address an application’s particular aspect and function, such as logging, data search, and more. Multiple such microservices come together to form one efficient application. This intuitive, functional division of an application offers several benefits. The client can use the user interface to generate requests. At the same time, one or more microservices are commissioned through the API gateway to perform the requested task.