Week-6 Homework Answers

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

Manual testing is done by MAN, that's why its called manual. But automated testing is done by helper softwares, tools and scripts. Automated testing is more reliable as computer doesn't make mistakes, and it is faster than manual testing. Moreover you can run tests as many times as you want while manual testing consumes time. But automated tests do not always report an understandable response, and manual testing can help developers more as they are actively witness the test steps and where error occurs.

2 - What does Assert class ?

Assert is a Java keyword used to define an assert statement. An assert statement is used to declare an expected boolean condition in a program. If the program is running with assertions enabled, then the condition is checked at runtime. If the condition is false, the Java runtime system throws an AssertionError .

3 - How can be tested 'private' methods ?

Private methods should not be tested separately. If you are trying to test something private, you should ask yourself, why is it private in the first place. Private methods are called by public methods that they are tied to. You can try to test its parent public method to see if it does the private method.

4 - What is Monolithic Architecture ?

Monolithic architecture is the commonly unified model for the design of a software program. Monolithic, in this context, means composed all in one piece. Monolithic software is designed to be self-contained. This means components of the program are interconnected and interdependent rather than loosely coupled as is the case with modular software programs. In a tightly-coupled architecture, each component and its associated components must be present in order for code to be executed or compiled.

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

- Tests Should Be Fast
- Tests Should Be Simple
- Test Shouldn’t Duplicate Implementation Logic
- Tests Should Be Readable
- Tests Should Be Deterministic
- Make Sure They’re Part of the Build Process
- Distinguish Between The Many Types of Test Doubles and Use Them Appropriately
- Adopt a Sound Naming Convention for Your Tests
- Don’t Couple Your Tests With Implementation Details

6 - Why does JUnit only report the first failure in a single 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. It reports failure on each test. Shared setup code is most natural when sharing between tests. This is a design decision that permeates JUnit, and when you decide to report multiple failures per test, you begin to fight against JUnit. This is not recommended. The resulting tests use JUnit's natural execution and reporting mechanism and, failure in one test does not affect the execution of the other tests. You generally want exactly one test to fail for any given bug, if you can manage it.

7 - What are the benefits and drawbacks of Microservices ?

Advantages of Microservices:

Microservices are self-contained, independent deployment module.

The cost of scaling is comparatively less than the monolithic architecture.

Microservices are independently manageable services. It can enable more and more services as the need arises. It minimizes the impact on existing service.

It is possible to change or upgrade each service individually rather than upgrading in the entire application.

Microservices allows us to develop an application which is organic (an application which latterly upgrades by adding more functions or modules) in nature.

It enables event streaming technology to enable easy integration in comparison to heavyweight interposes communication.

Microservices follows the single responsibility principle.

The demanding service can be deployed on multiple servers to enhance performance.

Less dependency and easy to test. Dynamic scaling. Faster release cycle.

Disadvantages:

Microservices has all the associated complexities of the distributed system.

There is a higher chance of failure during communication between different services.

Difficult to manage a large number of services.

The developer needs to solve the problem, such as network latency and load balancing.

Complex testing over a distributed environment.

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

In basics, Actuator brings production-ready features to our application. Monitoring our app, gathering metrics, understanding the traffic, or the state of our database become trivial with this dependency. The main benefit of this library is that we can get production-grade tools without having to actually implement these features ourselves. 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 ?

Microservices are hard to test. It's even more difficult when there are number of them. Developer should solve the problem on its own, it's hard if the developer doesn't know how each microsystem operates. In short it has a complex structure you have to know what you are doing, at least be familiar with the microservice you are using.

10 - How independent microservices communicate with each other?

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 (DDD) is a software design approach focusing on modelling software to match a domain according to input from that domain's experts. In terms of object-oriented programming it means that the structure and language of software code (class names, class methods, class variables) should match the business domain. For example, if a software processes loan applications, it might have classes like LoanApplication and Customer, and methods such as AcceptOffer and Withdraw. DDD connects the implementation to an evolving model.

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. Compared to server or machine virtualization approaches, however, containers do not contain operating system images. This makes them more lightweight and portable, with significantly less overhead. In larger application deployments, multiple containers may be deployed as one or more container clusters. Such clusters might be managed by a container orchestrator such as Kubernetes.

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 architectures have come into use along with Docker containers—a packaging and deployment construct. VM images have been used as the deployment mechanism of choice for many years. But containers are even more efficient than VMs, allowing the code (and required code libraries) to be deployed on any Linux system (or any OS that supports Docker containers). Containers are the perfect deployment vector for microservices. They can be launched in seconds, so they can be redeployed rapidly after failure or migration, and they can scale quickly to meet demands. Because containers are native to Linux, commodity hardware can be applied to vast farms of microservices in any data center, private cloud, or hybrid multicloud.