***Spring Boot***

1. **What is Spring Boot?**

* Spring Boot is an open-source Java-based framework used to create a micro-Service.
* It is used to build stand-alone and production ready spring applications that we can just run.
* We can get started with minimum configurations without the need for an entire Spring configuration setup.

1. **What is Spring?**

* Application Framework.
* Programming and configuration model.
* Infrastructure Support.

1. **Problems with Spring?**

* Huge Framework.
* Multiple setup steps.
* Multiple configuration steps.
* Multiple build and deploy steps.

1. **What Spring Boot Gives us?**

* Opinionated.
* Convention over configuration.
* Stand Alone.
* Production Ready.

1. **Maven?**

* Maven let us declare the dependencies we want into a single file called ***pom.xml***.
* ***pom.xml*** contains list of all our dependencies.

1. **Starting Spring Boot:**

* *SpringApplication.run(App.class, args);*
* Sets up default configuration.
* Starts Spring application context.
* Performs class path scan.
* Starts Tomcat server.

1. **What is a Controller?**

* It is a Java Class.
* Marked with Annotations.
* Has info about:
  + What URL access triggers it
  + What method to run when accessed

1. **What is a @RestController?**

* *Spring 4.0 introduced the* ***@RestController*** annotation in order to simplify the creation of RESTful web services.
* The ***@RestController*** annotation is used to define the RESTful web services. It serves JSON, XML and custom response.

1. **What is @RequestMapping?**

* The ***@RequestMapping*** maps only to the ***GET*** method by default.
* To map to other HTTP methods, we have to specify it in the annotation.

1. **What is Spring MVC?**

* A Spring MVC is a Java framework which is used to build web applications.
* It follows the **Model-View-Controller** design pattern.
* It implements all the basic features of a core spring framework like **Inversion of Control**, **Dependency Injection**.

1. **What are the HTTP Methods for RESTful Services?**

* **POST:** Create
* **GET:** Read
* **PUT:** Update/Replace
* **DELETE:** Delete

1. **What is @Service?**

* **@Service** annotation is used to indicate that a class belongs to service layer.
* **@Service** annotation is used with classes that provide some business functionalities.