Spring Boot:

***The SpringApplication.run():*** This method is a convenient entry point for Spring Boot applications and this method is a convenient way to bootstrap a Spring application.

It is part of the org.springframework.boot package.

\*\*\*\* The command-line arguments (String[] args):

These are the command-line arguments passed to the application. They can be used to customize the application behaviour at runtime.

SpringApplication.run() is called, it performs the following tasks:

1: Sets up the default configuration.

2: Starts the Spring application context.

3: Performs classpath scanning to find and register beans.

4: Starts an embedded web server if needed (for web applications).

Runs the application. (tomcat and jetty is embeded in spring boot while in spring we have to add it externally)

**Annotation used:**

@*SpringBootApplication*: This annotation combines @*Configuration*, @*EnableAutoConfiguration*, and @ComponentScan to enable the Spring Boot application. this annotation is used to enable various configurations, including component scanning, auto-configuration.

1. **@Configuration**:
   * Indicates that the class provides configuration to the Spring IoC (Inversion of Control) container. It is used to define and configure beans in the application context.
2. **@EnableAutoConfiguration**:
   * Enables Spring Boot's auto-configuration feature. Auto-configuration automatically configures the Spring application context based on the dependencies present in the classpath. It aims to reduce the amount of boilerplate configuration needed in a Spring application.
3. **@ComponentScan**:
   * Instructs Spring to scan and discover beans (components, services, repositories, etc.) within the specified package and its sub-packages. It's a way to tell Spring where to look for components to automatically register in the application context.

**\*\*\* Explain Spring Boot \*\*\***

Spring Boot is an open-source framework within the larger Spring Framework ecosystem that simplifies the development of Java-based applications, particularly web and microservices applications. It provides a set of conventions, defaults, and tools to streamline the development process, making it easier for developers to create production-ready applications with minimal configuration.

Key features and concepts of Spring Boot include:

1. **Convention over Configuration:**
   * Spring Boot follows the principle of "convention over configuration," which means that developers are provided with sensible defaults and conventions. This reduces the need for explicit configuration, allowing developers to focus more on writing business logic.
2. **Auto-Configuration:**
   * Spring Boot includes a powerful auto-configuration mechanism that automatically configures the application based on its dependencies. It examines the classpath, detects the libraries present, and configures the application accordingly. This feature simplifies the setup of various components like databases, messaging systems, and more.
3. **Embedded Web Server:**
   * Spring Boot comes with an embedded web server (Tomcat, Jetty, or Undertow) by default. This eliminates the need for developers to deploy applications to external web servers, making it easier to run and deploy standalone applications.
4. **Microservices Support:**
   * Spring Boot is well-suited for building microservices architectures. It includes features for creating RESTful APIs, handling configuration across services, and simplifying communication between microservices.
5. **Spring Boot Starters:**
   * Starters are a set of pre-configured dependencies that simplify the inclusion of common sets of functionalities, such as web applications, data access, messaging, etc. They reduce the amount of boilerplate configuration required for various tasks.
6. **Spring Boot Actuator:**
   * Actuator provides built-in production-ready features like health checks, metrics, and monitoring. It allows developers and operators to understand and monitor the state of the application.
7. **Spring Boot CLI (Command Line Interface):**
   * The CLI allows developers to write and run Spring Boot applications directly from the command line, making it convenient for quick prototyping and development.
8. **Spring Boot DevTools:**
   * DevTools provide tools for development, including automatic application restarts, live reloading of changes, and enhanced logging.
9. **Spring Initializr:**
   * The Spring Initializr is a web-based tool that simplifies the process of creating a new Spring Boot project. It allows developers to select dependencies, configure project settings, and download a fully configured project structure.