1) Inversion of control is design principle. It can be explained that If we think a class or service, we can give control of its dependencies and lifecycle into a container or a framework. In other words developers don't care responsibilities like dependencies etc, another structure do this for developers. Some examples of these responsibilities are creating objects of a class, knowing details of creating objects, managing lifecycle. Ioc help us to design testable, maintainable and extensible loose couple classes.
Dependency Injection is an implemantation method of IoC. Dependency Injection aims that a class gets rid of its dependency. By applying DI, we can provide that a class can act independently without its dependency objects. For DI, code should be written based on inteface.

2) Bean means object which is managed by Ioc container. Ioc container determines when and how created beans. Beans are reusable and have lifecycle.
Beans do what we want according to beans lifecycle. So we should determine scope for bean. There are 5 type bean scope; singleton, prototype, request, session, global session.

3) @SpringBootApplication is an annotation and indicate us class is a configuration class. @SpringBootApplication annotation contains these three annotations: @ Configuration, @EnableAutoConfiguration, @ComponentScan. @Configurations performs java based configurations. @Componentscan provides automatic scanning for components and @EnableAutoConfiguration carry out automatically default configurations.

4) Projects have common process like logging, transactions, caching etc. These common process are called cross-cutting concerns. AOP is about seperation of cross cutting concerns. Abstraction provides seperation of concerns in OOP. But it isn't enough. So, AOP helps OOP. AOP increases code readibility, reusablity by seperating concerns. We can seperate concerns via interceptor. Interceptors are structures which allow us to perform operations before and after a method is called.

5) Singleton is a design pattern. It provides creating class have only one instance. Singleton objects creates one times when it is called. We can use singleton design pattern in the places where new instances are created for each client and when we call the helper functions.

6) Spring Boot Actuator is used to manage and follow the application. Actuator have HTTP end points. Due to actuator, we can get information about our project. Also Spring boot actuator automatically activate features of applications. Pom file should add to Maven dependency code block to activate Spring Boot actuator.

7) Spring framework can apply all layer of a web application. Spring Boot creates stand-alone applications. Spring Boot offers embedded server like Tomcat,Jetty. Spring framework support xml and annotation configurations. Spring Boot don't need xml configurations. Spring boot aims to decrease lines of code. Spring framework doesn't support for an in-memory database. Spring Boot has in memory database like H2.

8) Version control is important for keep track of changes. It provides that every team member work with last version. It helps us to develop project rapidly. The version control system allows us to monitor and work together with our team members at the same workspace.

9) SOLİD is knew as standarts of OOP projects and consists of first letters these standarts. Single reponsibility principle means that a object should have only one responsibility. For example: Each method should do only its work, don't update multiple models same time. Open/Close principle means that Objects should be open to develop and be close to changes. Liskov's Substitution Principle means that inheritance class should use all of the property of inheritance objects.

Interface Segregation Principle means that more capabilities should not be added to an interface than necessary. Dependency Inversion Principle means that Top-level classes should not depend on lower-level classes and changes to subclasses should not affect superclasses.

10) RAD Model or Rapid Application Development model is a software development process based on prototyping without any specific planning. In RAD model, there is less attention paid to the planning and more priority is given to the development tasks. It targets at developing software in a short span of time.

11) Spring Boot Starters are dependency descriptors that can be added under the <dependencies> section in pom.xml. Using spring boot starter provides developers to decrease configuration time and increase productivity. Also it provides to manage POM easily.

12) A cache is a high-speed data storage layer that stores a temporary subset of data. Caching provides reuse of data which previously computed. Thanks to caching, database cost is reduced, application performance, reading efficiency is increased. We can enable caching in the Spring Boot application by using the annotation @EnableCaching.

13) Log is defined as record in programming. Logging is the process of recording application actions and state to a secondary interface. A large part of software developers lives are monitoring, troubleshooting and debugging. Logging makes this a much easier and smoother process.

14) Swagger is a technology that provides an essential contract standard for Rest API development and helper tools that function within this framework. Swagger provides convenience in API design, development, documentation and testing with the standards and tools it offers. Swagger can include to project at Maven by adding as dependency in pom.xml file.