1)What is spring

* Spring is used to make lightweight, loosely coupled enterprises applications in java.
* It provides pre-defined API for database operations, Restful web services, micro services, and security.
* We can develop distributed applications very fast. It is mainly for backend application development using java.

2) What are the features of Spring Framework

* Pre-defined templates (JDBC, HIBERNATE ETC.)
* Loose Coupling
* Light weight
* Easy to test. (Junit and Mockito)
* Powerful Abstraction
* Fast Development (Rapid Applications development RAD)

3) What is Dependency Injection (DI)

* It is a mechanism which is used to inject the dependency class into dependent class.
* Basically, when we write any program sometime its needed to create a object of some class, So dependency injection helped us to create object creation process, It will do that work automatically by own
* For achieving this functionality, we have only configure that particular dependency class to dependent class in Xml file or we can use annotation also like (@Autowired, @Qualifier,@Value,@Primary,@Inject)

4)What are the types of DI?

There are 5 types of Dependency Injections

* Constructor Injection
* Setter Injection
* Field Injection
* Circular Injection
* Lookup Method Injection

5)What is IOC in spring?

* The full form of Ioc is inversion of Control.
* In Spring, the IOC container (also known as the application context) takes control of managing objects and their dependencies. It achieves this by leveraging the concept of Dependency Injection (DI), where the required dependencies of a class are "injected" into it by the container.
* The IOC container in Spring identifies the dependencies of a class based on their configurations and automatically resolves them. This allows the application code to focus on implementing the business logic without worrying about creating and managing dependencies.
* The benefits of IOC in Spring include:

1. Loose coupling: With IOC, the classes are decoupled from their dependencies, making it easier to change and maintain the codebase.
2. Testability: IOC enables easier unit testing by allowing dependencies to be mocked or replaced with test-specific implementations.
3. Reusability: IOC promotes the reuse of components by separating their creation and management from the application logic.
4. Modular development: The IOC approach encourages modular development, as components can be easily replaced or extended without impacting the entire application.

* Overall, IOC in Spring helps in achieving a more flexible, maintainable, and scalable application architecture by managing the lifecycle of objects and their dependencies.