1. Do proper research on Why use Spring framework, Dependency, Dependency injection, Inversion of control, IoC container, ApplicationContext, BeanFactory, AbstractApplicationContext, Configuration metadata, Bean, Constructor injection, Setter injection, Autowiring, Modes of autowiring, Bean scopes, Singleton, Prototype, Spring bean life cycle, limitations of autowiring, List out all the annotations that can be used in Spring Core and when they are used?
2. Assignment to inject objects of an interface → Create a class Earth which has some attributes and a dependency Creature which will be an interface. Creature interface can have several implementing classes like HumanBeing, Lion, Peacock, GoldFish, etc. Your task is to be able to inject a Creature type object (i.e. object of implementing classes - HumanBeing, Lion, Peacock, GoldFish, etc.) in the World object.

   —----

   Hint:

   —--------------------

```
class Earth {
      private double planetRadius;
      private double waterOnPlanetPercentage;

      /* Dependency injection for interface Creature type attribute */
      private Creature creature;
      // code...
}
```

   —------

```
interface Creature {
      String creatureName();
      String creatureType();
      // code...
}
```

   —------

   Implementing classes of Creature interface can be HumanBeing, Lion, GoldFish, etc.

   —--------------------
3. We have learned how to inject Collection types such as List, Set and Map. Now your task is to inject Properties type objects in the depending object.

   —----

   Hint: Properties is also a Collection type data structure and injection of Properties type objects is very similar to other Collection types that we have already seen.

   —--------------------
4. Assignment for the usage of @Import → Create classes that represent food such as Biryani, Idli, FriedRice, etc and also create classes that represent drinks such as CocaCola, MilkShake, AppleJuice, etc. By using @Configuration annotation create configuration class for food example: FoodConfig.java and also create configuration class for drinks example: DrinksConfig.java. In respective configuration files use @Bean to create methods that will give the bean object

i.e. In FoodConfig.java we can have methods such as getBiryani(), getIdli(), getFriedRice(), etc. and similarly in DrinksConfig.java we can have respective methods that can provide bean objects. Now make use of @Import to combine both the configuration files in a configuration file called AllConfig.java. And then perform dependency injection using the AllConfig.java configuration file.

5. Assignment to use pure annotation based configuration and perform dependency injection → Create a class Child which has attributes such as name, age, gender, standard, hobbies (List type), subjectsStudying (Set type), friendsAndGender (Map type <String name, String gender>) and Pet (Interface). Pet interface can have multiple implementing classes such as Dog, Car, Cow, etc. Perform dependency injection for all the attributes by using pure annotation based configuration.

   —----

   Hint:

   —--------------------

```
class Child {
      private String name;
      private int age;
      private String gender;
      private int standard;
      private List<String> hobbies;
      private Set<String> subjectsStudying;
      private Map<String, String> friendsAndGender;
      private Pet pet;
      // code...
}
```

   —------

```
interface Pet {
      String petType();
      String petName();
      // code...
}
```

   —------

   Pet interface can have multiple implementing classes such as Dog, Fish, Cat, Cow, etc.

   —--------------------