Week 1 spring assignments:

Deadline 11 March, 11:59 pm)

Subsequent assignments will be provided before 11 march  
All assignment works must be committed to github created with corresponding office emails

Assignment 1:

**Inversion of Control with XML Configuration**

1. Define a new implementation for the Coach interface. You can use whatever sport you would like.
2. Reference your new implementation in the Spring config file.
3. Test your application to verify you are retrieving information from your new Coach implementation.

Assignment 2:

**Dependency Injection with XML Configuration**

1. Define a new implementation for the FortuneService.

    a. When the getFortune() method is called it should return a random fortune from the array.

    b. Your fortune service should define three fortunes in an array.

2. Inject your new dependency into your Coach implementation.

3. Test your application to verify you are retrieving random fortunes.

Assignment 3:

**Bean Scopes with XML Configuration**

1. Add bean scopes to your new Coach implementation that you created in one of the previous activities.

2. Test singleton scope and prototype scope as I did in the videos.

3. Verify that the bean scopes are being applied as desired.

Assignment 4:

**Inversion of Control with Java Annotations**

1. Define a new Coach implementation using Annotations
2. Reference the new coach implementation in your main application.
3. Test your application to verify you are getting information from your new coach implementation.

Assignment 5:

**Dependency Injection with Annotations**

1. Define a new implementation for the FortuneService.

* Your fortune service should read the fortunes from a file.
* The fortune service should load the fortunes into an array
* When the getFortune() method is called it would return a random fortune from the array.

2. Inject your new dependency into your Coach implementation

3. Test your application to verify you are getting random fortunes based on your fortunes file.

Assignment 6:

**Bean Scopes with Annotations**

1. Modify your file based fortune service to make use of the @PostConstruct annotation

2. In the @PostConstruct code, read the fortunes from the file system

3. Test your application and verify your @PostConstruct code is being executed.   
Hint, add some println statements.

Assignment 7:

**Inversion of Control and Dependency Injection with Java Code**

*Note: In this practice activity, do not use component scanning.*

1. Create a new Coach implementation.
2. Create a new fortune service implementation (return a single hard-coded fortune)
3. Create a Java configuration class to perform IoC and DI with those classes.
4. Develop a main application to retrieve your coach implementation.
5. Test your application and verify your coach implementation is wired up as desired.