```java
package com.study.DemoAnnotations;

public interface FortuneService {
    public String getFortune();
}

//////////////////////////////////////////////////////////////////////////////

package com.study.DemoAnnotations;

import org.springframework.stereotype.Component;

@Component
public class HappyFortuneService implements FortuneService {

    @Override
    public String getFortune() {
        return "Its a happy day today";
    }
}

//////////////////////////////////////////////////////////////////////////////
package com.study.DemoAnnotations;

import java.util.Random;

import org.springframework.stereotype.Component;

@Component
public class RandomFortuneService implements FortuneService {

    String[] messages = { "Beware of sheep in wolf clothing",
            "Diligence is the mother of good work",
            "Journey is the reward" };

    private Random random = new Random();

    @Override
    public String getFortune() {
        int index = random.nextInt(messages.length);
        return messages[index];
    }
}

//////////////////////////////////////////////////////////////////////////////

package com.study.DemoAnnotations;

public interface Coach {
    public String getDailyWorkout();

    public String getDailyFortune();
}

//////////////////////////////////////////////////////////////////////////////

package com.study.DemoAnnotations;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

@Component
public class TennisCoach implements Coach {

    private FortuneService fortuneService;

    // using constructor injection
    @Autowired
    public TennisCoach(FortuneService fortuneService) {
        super();
        this.fortuneService = fortuneService;
    }

    @Override
    public String getDailyWorkout() {
        return "Practice your backhand volley";
    }

    @Override
    public String getDailyFortune() {
        return fortuneService.getFortune();
    }
}

//////////////////////////////////////////////////////////////////////////////


package com.study.DemoAnnotations;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class AnnotationsDemoApp {

    public static void main(String[] args) {
        // load the spring configuration file
                ClassPathXmlApplicationContext context = new
                    ClassPathXmlApplicationContext("annotations-applicationContext.xml");

                // retrieve the bean from the spring container
                Coach theCoach = context.getBean("tennisCoach", Coach.class);
                // call methods on bean
                System.out.println(theCoach.getDailyWorkout());

                System.out.println(theCoach.getDailyFortune());

                // close the context
                context.close();
    }
}

//////////////////////////////////////////////////////////////////////////////

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd">

    <!-- add entry to enable component scanning -->
    <context:component-scan base-package="com.study.DemoAnnotations" />

</beans>

//////////////////////////////////////////////////////////////////////////////
/** Modified class for Auto Wiring by Setter methods or any method **/

package com.study.DemoAnnotations;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

@Component
public class TennisCoach implements Coach {

    private FortuneService fortuneService;
```

```java

    /*
     * public TennisCoach(FortuneService fortuneService) { super();
     * this.fortuneService = fortuneService; }
     */

    public TennisCoach() {
        super();
        System.out.println("Inside the default constructor of TennisCoach");
    }

    // setter function for Autowiring ..... actually it can be any method name
    @Autowired
    public void setFortuneService(FortuneService fortuneService) {
        System.out.println("Inside the set method setFortuneService of TennisCoach ");
        this.fortuneService = fortuneService;
    }

    @Override
    public String getDailyWorkout() {
        return "Practice your backhand volley";
    }

    @Override
    public String getDailyFortune() {
        return fortuneService.getFortune();
    }
}

//////////////////////////////////////////////////////////////////////////////

package com.study.DemoAnnotations;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

@Component
public class TennisCoach implements Coach {

    @Autowired
    private FortuneService fortuneService;

    /*
     * public TennisCoach(FortuneService fortuneService) { super();
     * this.fortuneService = fortuneService; }
     */

    public TennisCoach() {
        super();
        System.out.println("Inside the default constructor of TennisCoach");
    }

    /*
     * // setter function for Autowiring ..... actually it can be any method name
     *
     * @Autowired public void setFortuneService(FortuneService fortuneService) {
     * System.out.println("Inside the set method setFortuneService of TennisCoach "
     * ); this.fortuneService = fortuneService; }
     */

    @Override
    public String getDailyWorkout() {
        return "Practice your backhand volley";
    }

    @Override
    public String getDailyFortune() {
        return fortuneService.getFortune();
    }
```

```java
}

//////////////////////////////////////////////////////////////////////////////

@Autowired
public TennisCoach(@Qualifier("randomFortuneService") FortuneService theFortuneService) {

    System.out.println(">> TennisCoach: inside constructor using @autowired and @qualifier");

    fortuneService = theFortuneService;
}
//////////////////////////////////////////////////////////////////////////////


// add support to resolve ${...} properties
@Bean
public static PropertySourcesPlaceholderConfigurer
                propertySourcesPlaceHolderConfigurer() {

    return new PropertySourcesPlaceholderConfigurer();
}
```

```java
package com.study.config;

import org.springframework.context.annotation.Bean;
/*import org.springframework.context.annotation.ComponentScan;*/
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.PropertySource;

@Configuration
/* @ComponentScan("com.study.config") */
@PropertySource("classpath:sport.properties")
public class SportConfig {

    // define bean for a SadFortuneService, method name is the actual bean id
    @Bean
    public FortuneService sadFortuneService() {
        return new SadFortuneService();
    }

    @Bean
    public Coach swimCoach() {
        return new SwimCoach(sadFortuneService());
    }
}
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd">

    <!-- add entry to enable component scanning -->
    <context:component-scan base-package="com.study.springdemo" />

    <!--  Load the properties file -->
    <context:property-placeholder location="classpath:sport.properties"/>

    <bean id="myLoggerConfig" class="com.study.springdemo.MyLoggerConfig" init-method="initLogger">
        <property name="rootLoggerLevel" value="FINE" />
        <property name="printedLoggerLevel" value="FINE"/>
    </bean>
    <!--  define the dependency -->
    <bean id="myFortuneService"
        class="com.study.springdemo.HappyFortuneService">
    </bean>

    <bean id="myCoach"
        class="com.study.springdemo.TrackCoach">

        <!--  setup constructor injection -->
        <constructor-arg ref="myFortuneService"></constructor-arg>
    </bean>

    <bean id="myCricketCoach"
        class="com.study.springdemo.CircketCoach" scope="singleton">

        <!--  setup setter injection -->
        <property name="fortuneService" ref="myFortuneService" />

        <!--  setup setter injection for literal values -->
        <property name="emailAddress" value="${foo.email}" />
        <property name="team" value="${foo.team}" />

    </bean>
</beans>
```