**Inversion Of Control**

1. The approach of outsourcing the construction (creation) and management of objects.
2. Outsourcing is handled by the object factory
3. Always code to interface, which means create interfaces wherever possible.
4. Interface says what is available not how is it available.

**Spring Container**

1. Inversion of Control – We invoke and create objects using the config file.
2. Dependency Injection – We inject the objects dependencies after the invocation.

**Configuring Spring Container**

1. XML configuration file
2. Java Annotations
3. Java Source Code

**Spring Development Process**

1. Configure your Spring Beans
2. Create a Spring Container
3. Retrieve Beans from Spring Container

**Configuring Spring Beans**



1. Id is used to fetch the bean

**Creating a Spring Container**

1. Spring container is generically known as **ApplicationContext**.
2. 

**Retrieve Beans from Container**

1. A screenshot of a cell phone
   
   Description automatically generated
2. Here Coach.class is the interface and com.luv2code.springdemo.BaseballCoach is the implementation of that interface class.

**Spring Bean**

1. A spring bean is simply a Java Object
2. When java objects are created from the spring container they are refereed to as Spring Beans.

**Implementation of IoC**

A screenshot of a cell phone

Description automatically generated

**Logging Messages on Spring 5**

1. Create a bean to configure the parent logger and console handler

* This sets up the logging events to fine you could make it finer.
* This class also has an init method to handle actual configuration, init is executed after the bean is created and dependencies injected.

1. Configure the bean in the Spring XML config file

* Make sure to add this bean as the first bean on the XL file
* A screenshot of a cell phone
  
  Description automatically generated