

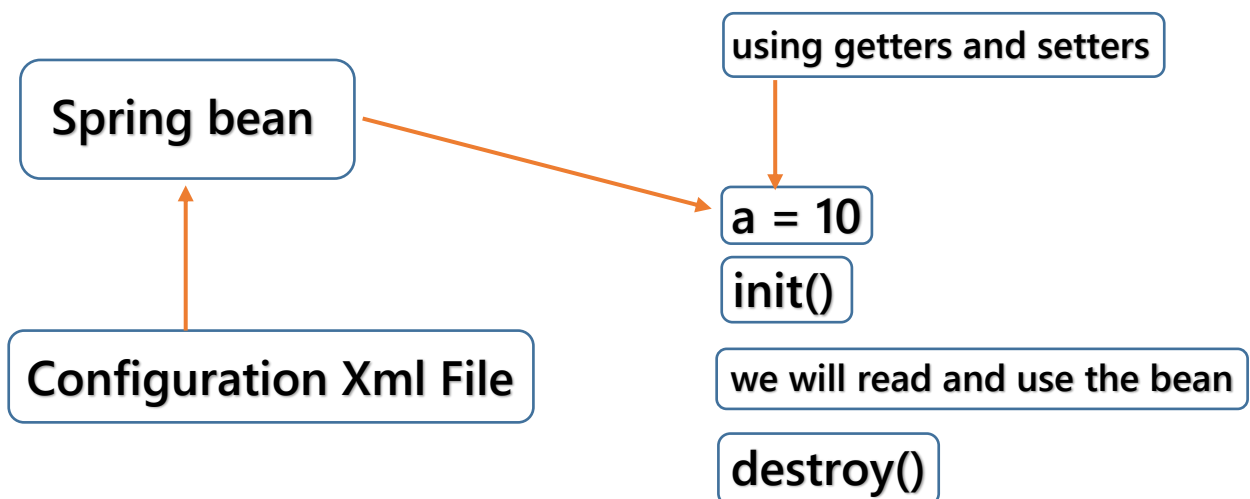
## Life Cycle Methods of Spring Beans

The lifecycle of any object means when & how it is born, how it behaves throughout its life, and when & how it dies. Similarly, the bean life cycle refers to when & how the bean is instantiated, what action it performs until it lives, and when & how it is destroyed.

Spring provides two important methods to every bean:-

1. `init()` – For initializing the code
2. `destroy()` – For cleaning the code/destroy the objects of spring

Bean life cycle is managed by the spring container. When we run the program then, first of all, the spring container gets started. After that, the container creates the instance of a bean as per the request, and then dependencies are injected. And finally, the bean is destroyed when the spring container is closed. Therefore, if we want to execute some code on the bean instantiation and just after closing the spring container, then we can write that code inside the custom `init()` method and the `destroy()` method.



There are three approaches to configure Spring Bean Life Cycle methods

- Using XML (also called Declarative Approach)
- Using Spring Interfaces (also called Programmatic Approach)
- Using Annotations.

#### 4. Java 9+

Note that both the `@PostConstruct` and `@PreDestroy` annotations are part of Java EE. Since Java EE was deprecated in Java 9, and removed in Java 11, we have to add an additional dependency to use these annotations:

```
<dependency>
  <groupId>javax.annotation</groupId>
  <artifactId>javax.annotation-api</artifactId>
  <version>1.3.2</version>
</dependency>
```

