Bean Factory:-

BeanFactory is the interface (or) the core container in the Spring framework

-> Beans acts as components

-> Basic functionality is dependency injection and bean management and also managing bean factory lifecycle.

-> Beans are acts as lightweight components.

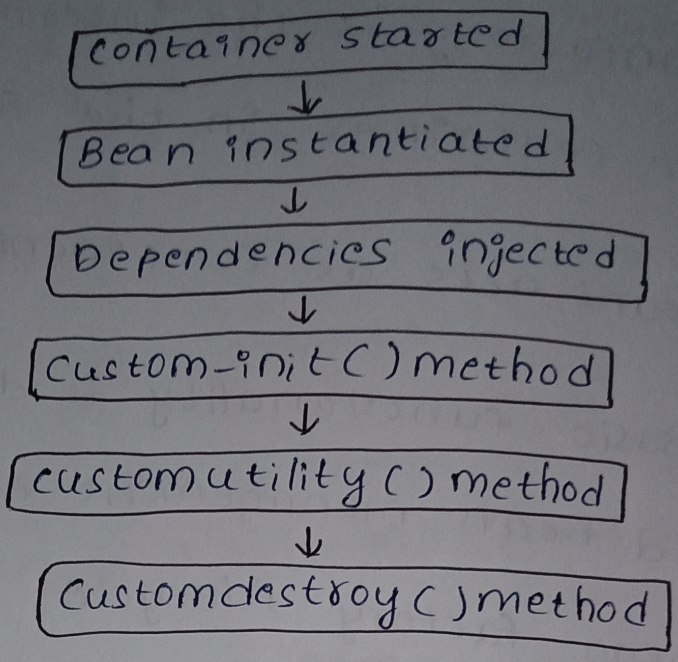
Beans are usually defined in:

-> xML files

-> using annotations
(like @ component, @ Bean)

-> Java-based configuration.

Lifecycle of Bean factory:-



(i) **Container started():**

Syntax:

```
Public className() {
    // initialization code
}
```

Example:-

```
Public class MyBean {
  Public MyBean() {
System.out.println("Bean object created
                    container started!");
  }
}
```

(ii) Dependencies injected:

Syntax:

```
<bean id="beanId" class="className">
  <property name="properyName"
  value="value" />

</bean>
```

Example:

```
Public class student
{
private string name;
public void setName (string name)
{
    this.name = name;
}
Public string display()
{
    return name;
}
}
```

(iii) Custom-init():

To initialize resources (e.g. database, connections, file handlers)

Syntax:

```
Public void methodname()
{
    statements;
}
```

Example:

```
Public void custominit()
{
system.out.println(Custom method called);
}
```

(v) customutility():

It is a normal java method defined inside a spring bean class to perform some bussiness logic, computation or service operation.

Syntax:

```
Public returntype methodname()
{
 return result;
}
```

Example:

```
Public Void customutility().
{
 return message;
}
```

(vi) customdestroy():

The custom destroy method is used to clean up resources (or) perform shutdown operations before a bean is destroyed by the spring container.

Syntax:

```
Public Void methodname()
{
  cleanup statements;
}
```

Example:

```
Public Void destroy()
{
 cleanup statements;
}
```

Advantages of BeanFactory:

1) Loose coupling

2) Reusability

3) Easy maintenance

4) Lifecycle Management

5) Dependency injection

6) Lightweight component

Disadvantages of BeanFactory:

1) No built-in event handling
2) No internationalization
3) No automatic bean preloading
4) Lacks AOP support
5) More manual effort for resource management

Applications of BeanFactory:

1) small spring based java applications
2) IoT (or) mobile apps with limited memory (lazy loading helps)
3) Testing environments where you load only specific beans
4) used in testing or small scale apps
5) changing a database connection without modifying java code