**What are annotations?**

Used to provide supplement information about a program. It starts with '@' and do not change the action of compiled program. it helps to associate metadata(information) to program elements. they can change the way a program is treated by compiler.

**What are the categories of Annotations?**

There are three categories:

1. **Marker Annotations:**

The only purpose is to mark a declaration. It contains no members and do not consist any data. thus its presence as annotation is sufficient. Since marker inteface contains no members simply determine whether it is present or absent is sufficient.

ex: @Override.@Entity

**2. Single Value Annotations:**

These annotations contains only one member and allow a short hand form of specifying the value of member.

ex: @TestAnnotation("testing"),@Retention(RetentionPolicy.RUNTIME)

**3. Full Annotations:**

These annotations consist of multiple data members name, value, pairs.

ex: @Table(name="sample", schema="sample")

**How annotations work internally?**

**First it checks for retention policy**

a. main distinction between kinds of annotation is whether they are used at compile time and then discarded like @override or placed in the compiled class available at runtime like @Component.

b. This determines by retention policy of annotation. if you are writing your own annotation, you'd need to decide whether the annotation is helpful at runtime or only at compile time.

c. When compiling code with annotations, compiler sees the annotation just like it sees other modifiers on source elements.

d. When it encounters an annotation, it runs an annotation processor, which is like a plug-in class.

e.  It uses the Reflection API to make sure it can find a match for the method signature in one of the superclasses and uses the Messager to cause a compile error if it can't.

**What are Retention Policies for Annotation?**

* RetentionPolicy.SOURCE retains an annotation only in the source file and discards it during compilation.

Eg: @Override @Suppresswarning @Deprecataed

* RetentionPolicy.CLASS stores the annotation in the .class file but does not make it available during runtime.
* RetentionPolicy.RUNTIME stores the annotation in the .class file and also makes it available during runtime.

Eg:@Configuration,@Autowired