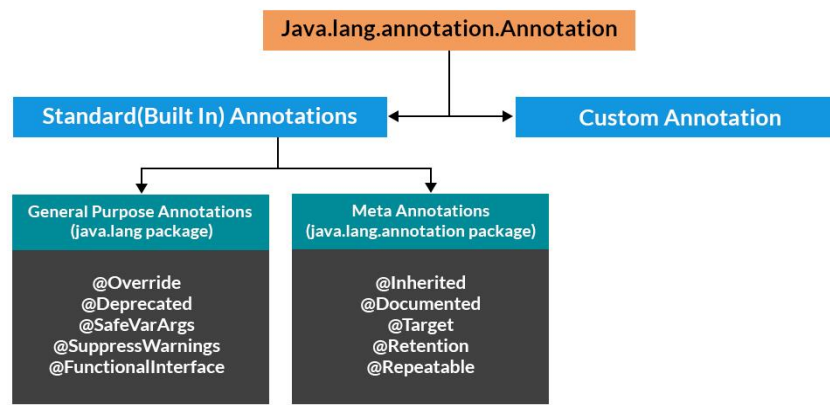


Annotations in Java

Annotations are used to provide supplemental information about a program.

- Annotations start with '@'.
- Annotations do not change the action of a compiled program.
- Annotations help to associate *metadata* (information) to the program elements i.e. instance variables, constructors, methods, classes, etc.
- Annotations are not pure comments as they can change the way a program is treated by the compiler. See below code for example.
- Annotations basically are used to provide additional information, so could be an alternative to XML and Java marker interfaces.



Example: @Override

```
import java.lang.annotation.*;
class Animal
{
    void eatSomething()
    {System.out.println("eating something");}
}

class Dog extends Animal{
    @Override
    void eatsomething()
    {System.out.println("eating foods");}
}

class Test{
    public static void main(String args[]){
        Animal a=new Dog(); a.eatSomething();
    }
}
```

```
C:\Windows\System32\cmd.exe
C:\Program Files\Java\jdk-11.0.12\bin\Manish>javac Test.j
ava
Test.java:8: error: method does not override or implement
a method from a supertype
@Override
^
1 error
```

Example: @SuppressWarnings

```
import java.util.*;
class Test
{
    //@SuppressWarnings("unchecked")
    public static void main(String args[])
    {
        ArrayList list=new ArrayList();
        list.add("Ankit");
        list.add("Abhishek");
        list.add("Tushar");

        for(Object obj:list)
        System.out.println(obj);
    }
}
```

```
C:\Windows\System32\cmd.exe
C:\Program Files\Java\jdk-11.0.12\bin\Manish>javac Test.j
ava
Note: Test.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
```

Demonstrate User-defined Annotations:

```
import java.lang.annotation.Documented;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;

// User-defined annotation
@Documented
@Retention(RetentionPolicy.RUNTIME)

@interface TestAnnotation
{
    String Developer() default "Suyash";
    String Expirydate();
} // will be retained at runtime

// Driver class that uses @TestAnnotation
public class Test
{
    @TestAnnotation(Developer="Suyash", Expirydate="01-10-2024")
    void fun1()
    {
        System.out.println("Test method 1");
    }

    @TestAnnotation(Developer="Param", Expirydate="01-11-2024")
    void fun2()
    {
        System.out.println("Test method 2");
    }

    public static void main(String args[])
    {
        System.out.println("Great to learn Java Annotation.");
    }
}
```