

override, SuppressWarnings, deprecated

marker annotations



Annotations :

→ only this in core Java.

There are total 15 annotations  
but only 3 in core java.

① @Override :

```
class Test {  
    void showTest() {  
        s.o.p("Show of Test");  
    }  
}
```

```
class ChildTest extends Test {  
    @Override  
    void showTest() {  
        s.o.p("Show of Child Test");  
    }  
}
```

```
class TestDemo {  
    public static void main(String[] args) {  
        ChildTest ct = new ChildTest();  
        ct.showTest();  
    }  
}
```

This will generate compile time error as showTest() is not  
an overridden method.

if annotation was not present the program would have  
run successfully.

only for ~~unchecked~~ warnings

```
(2) @SuppressWarnings {  
import java.util.*;  
class TestSuppruss {
```

```
@SuppressWarnings("unchecked");  
public static void main(String[] args) {  
    ArrayListStringlist = new ArrayList<String>();  
    list.add("amit");  
    list.add("sunil");  
    list.add("ajay");  
    for (Object obj : list)  
        S.O.P(obj);  
}  
}
```

(3) @Deprecated : → for methods which we are going to remove in future ~~code~~ version.  
eg. finalize was deprecated in 1.2  
" " removed in 1.9

```
class A {  
    void m() {  
        S.O.P("non-Deprecated function");  
    }  
    @Deprecated  
    void n() {  
        S.O.P("Deprecated function");  
    }  
}  
  
class DepDemo {  
    public static void main(String[] args) {  
        A obj = new A();  
    }  
}
```



Suspend () for Thread was  
removed in 1.6 version.  
before it was deprecated.

DATE \_\_\_\_\_  
PAGE \_\_\_\_\_

```
obj.h();
```

```
}
```

```
}
```

This program will run but generate ~~error~~ warning for use of deprecated method.