Wednesday, October 9, 2024 9:54 AM

Covariant return type

-> covariant return type means if method return type is non-primitive data member then we can return same or it's sub class object.

A.java

```java
package com.cjc1;

public class A {

}
```

B.java

```java
package com.cjc1;

public class B extends A {

}
```

C.java

```java
package com.cjc1;

public class C extends B{

}
```

Test.java

```java
    public A m1() {
        A  a = new A();
        B  b = new B();
        C  c  = new C();
        //return a;  // we can return a
        //return b;  // we can return b
        return c; // we can return c
    }
    public B m2() {
        A  a = new A();
        B  b = new B();
        C  c  = new C();

        //return a; // we can't return a
        //return b;  // we can return b
        return c; // we can return c
    }
    public  C  m3() {
        A  a = new A();
        B  b = new B();
        C  c  = new C();

        //return a; // we can't return a
        //return b;  // we can't return b
        return c; // we can return c
    }
```

What is non-primitive type casting ?

-> conversion of one data member into another data member known as type casting
-> there are two types of type casting is present upcasting and down casting

What is upcasting ?

-> conversion of lower data member (B) into higher data member (A) known as upcasting
-> for upcasting we don't need to perform any changes in code explicitly

Test.java

```
1  package com.cjc1;
2
3  public class Test {
4
5      public B m1() {
6          B b = new B();
7          return b;
8      }
9
10     public static void main(String[] args) {
11
12         Test t = new Test();
13
14         A  a = t.m1();
15
16         System.out.println(a.x);
17     }
18 }
19
```

What is down casting ?

-> conversion of higher data member (A) into lower data member known as down casting

-> to perform down casting we need to perform change in code explicitly

Test.java

```
1  package com.cjc1;
2
3  public class Test {
4
5      public A m1() {
6
7          B b = new B();
8
9          return b;
10     }
11
12     public static void main(String[] args) {
13
14         Test t = new Test();
15
16         B b = (B) t.m1();
17
18          System.out.println(b.y);
19     }
20 }
21
```

Note* type casting cannot happen in between classes that has no parent and child relationship with each other

Object example

Test.java

```java
package com.cjc1;

public class Test {

    public Object m1() {

        return "ABC";
    }

    public static void main(String[] args) {
        System.out.println("main----start");

        Test t = new Test();
        String i = (String) t.m1();

        System.out.println("main---ends");
    }
}
```