is-a relationship
Parent Child

Reusability ->

Child extends Parent{

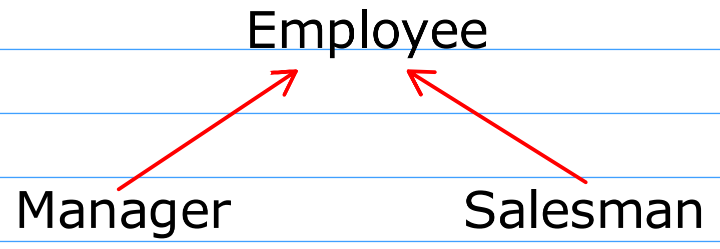
}

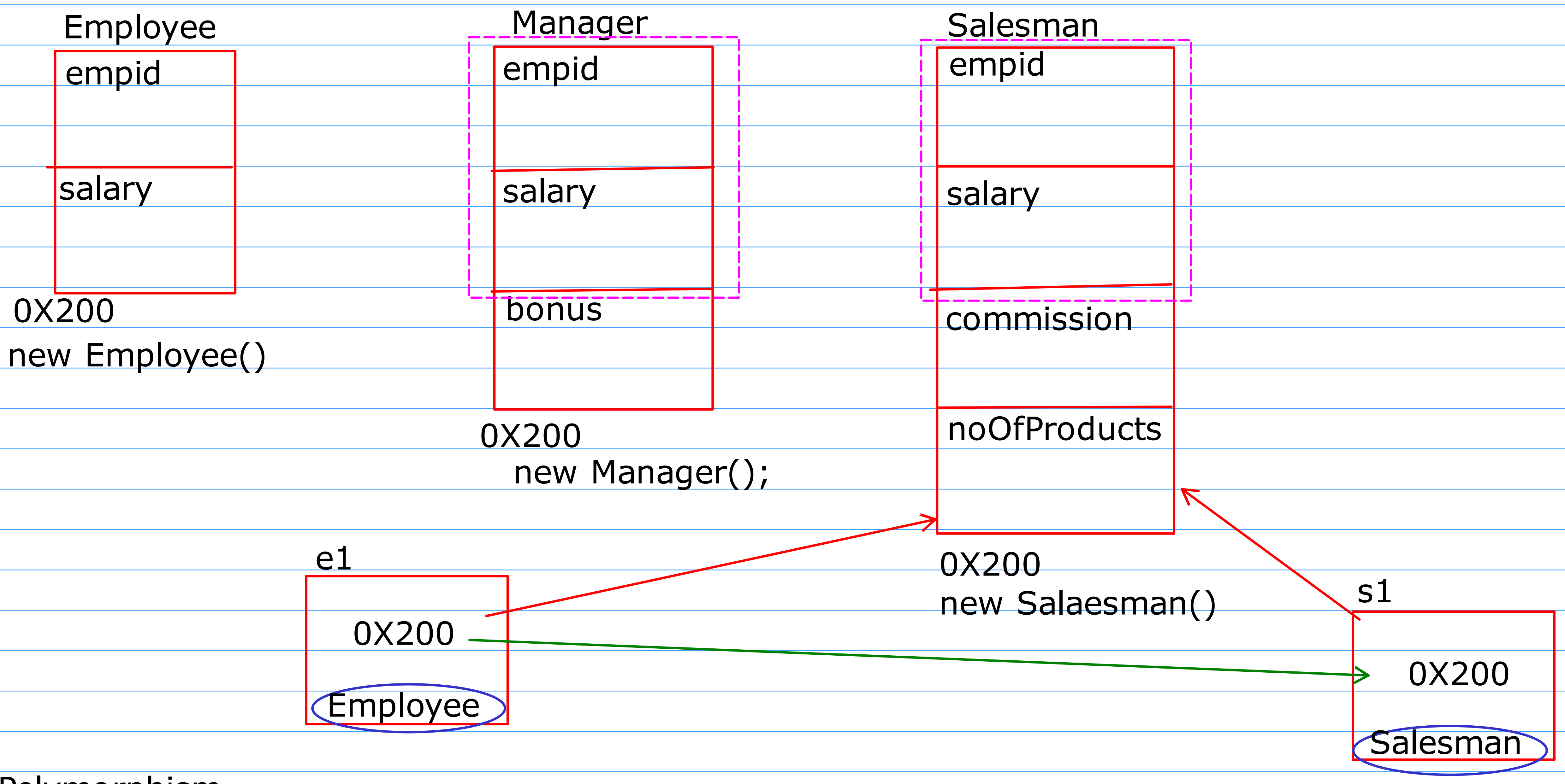
Method Overriding

- Its an example of RunTime Polymorphism

Rules of Method Overriding

Employee -> Manager, Salesman





Polymorphism

- One entity that takes multiple forms
- types of polymorphism

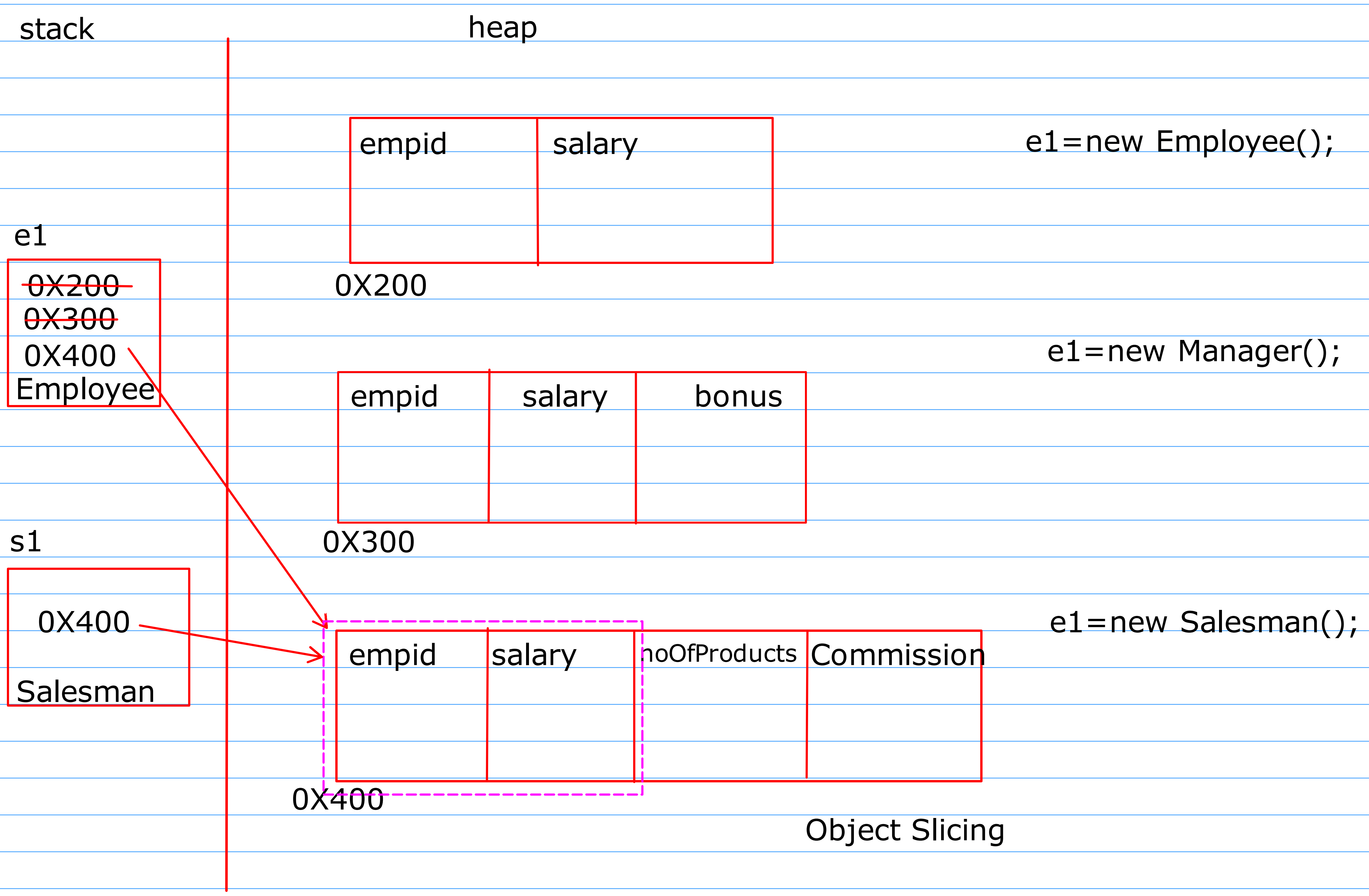
1. CompileTime
   - Method Overloading (name of method-> signature is differnt)
2. RunTime
   - Method Overriding (name of method -> signature is same)

Upcasting

- Storing the object of subclass in the super class reference
- When upcasting is done the super class reference can only call the members of the super class inherited inside the subclass, this is called as Object slicing

Downcasting

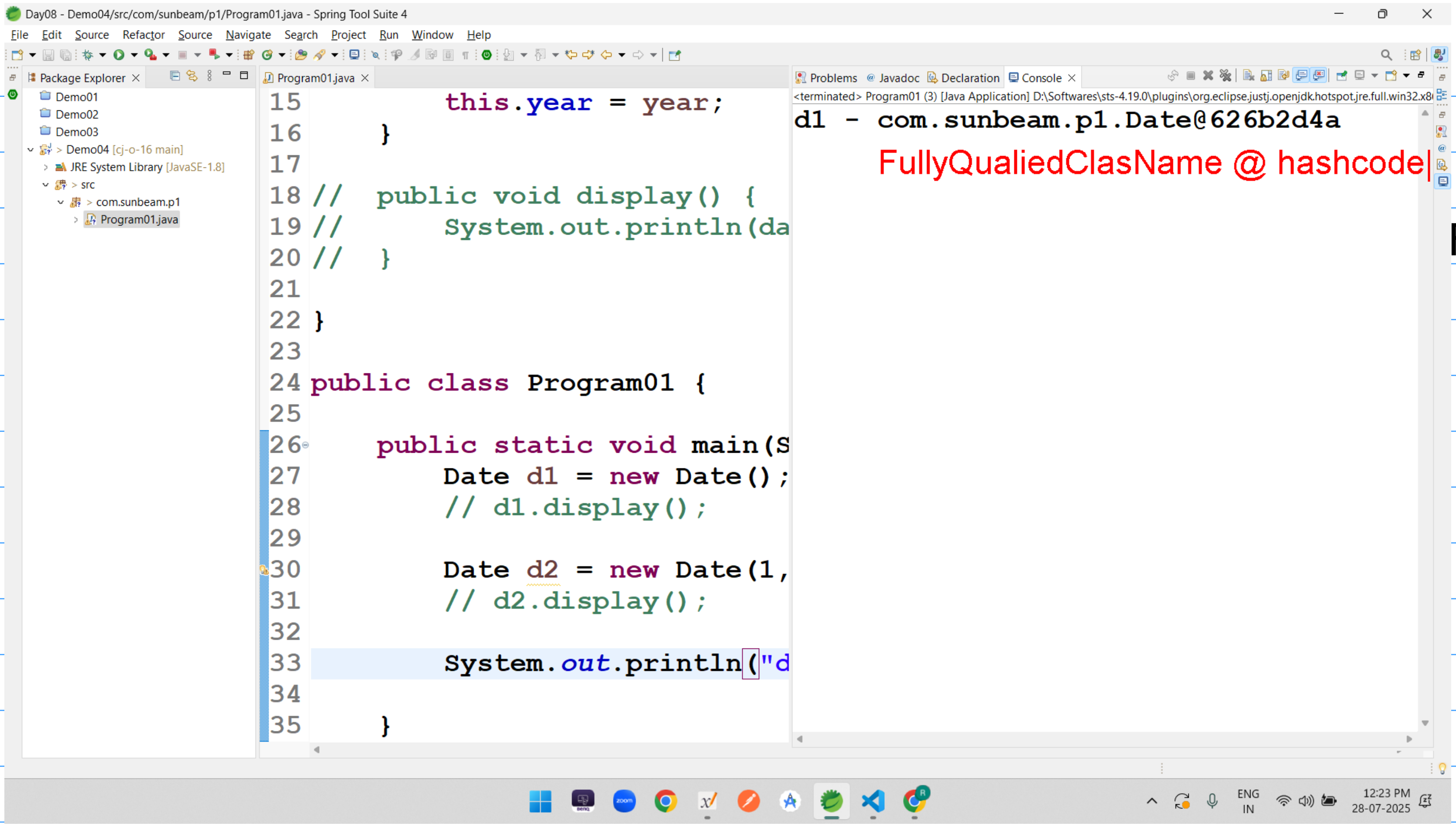
- Converting the reference of super class into the subclass reference
- While downcasting explicit typecasting is mandatory



Emp ref ->emp,manager,salesman (emp obj is present in all these classes)
Salesman ref -> salesman

final

- variable
- field (field/Object initializer or the ctor)
- method (100% complete) cannot override such methods
- class (100 % complete) cannot be extended


Day08 - Demo04/src/com/sunbeam/p1/Program01.java - Spring Tool Suite 4
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer
Demo01
Demo02
Demo03
Demo04 [cj-o-16 main]
JRE System Library [JavaSE-1.8]
src
com.sunbeam.p1
Program01.java
Problems
Javadoc
Declaration
Console
ENG
IN
12:23 PM
28-07-2025

```
        this.year = year;
    }

//  public void display() {
//      System.out.println(da
//  }

}

public class Program01 {

    public static void main(S
        Date d1 = new Date();
        // d1.display();

        Date d2 = new Date(1,
        // d2.display();

        System.out.println("d

    }
```

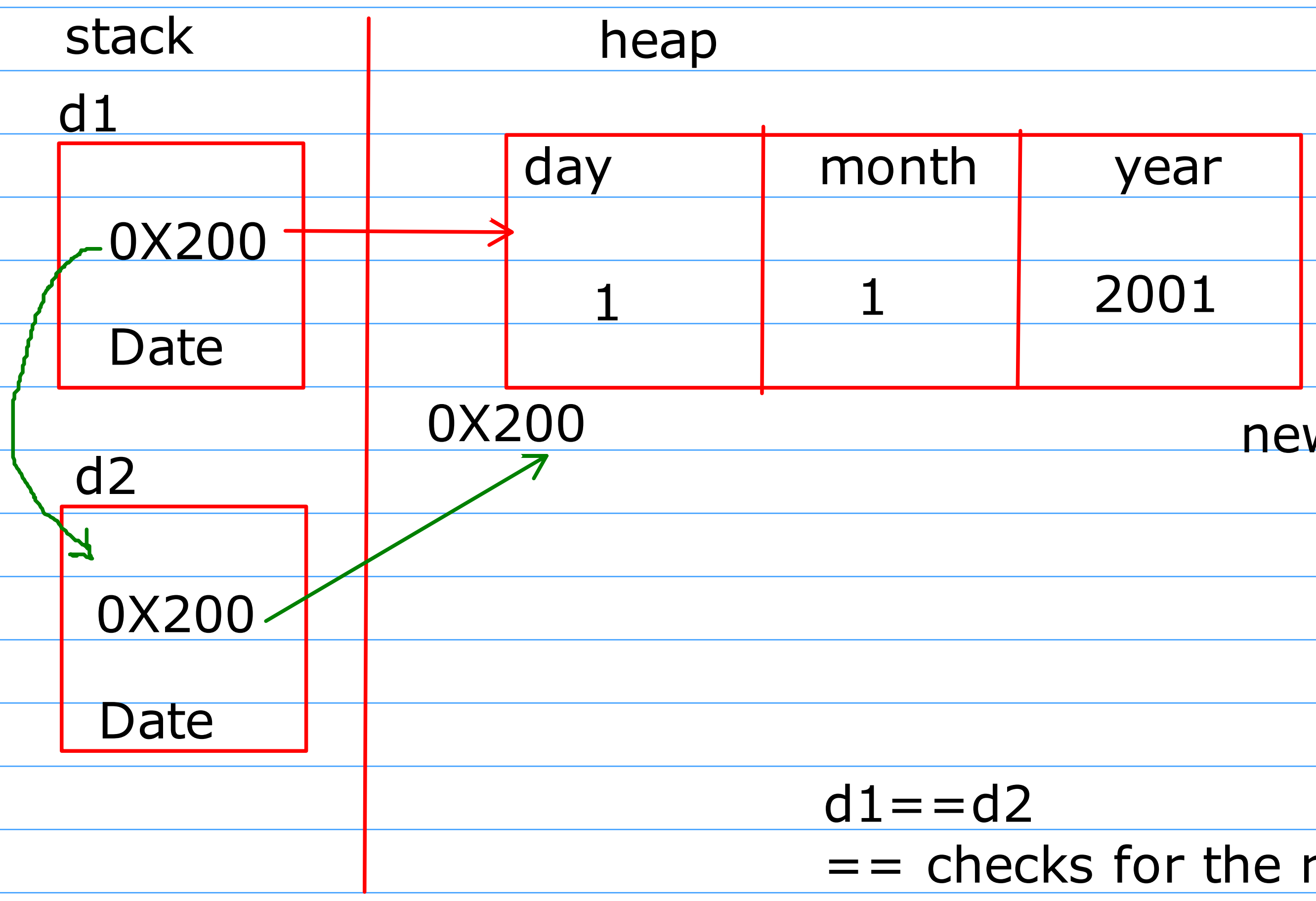
```
<terminated> Program01 (3) [Java Application] D:\Softwares\sts-4.19.0\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x8
d1 - com.sunbeam.p1.Date@626b2d4a
```

FullyQualiedClasName @ hashcode

Object class toString() represents

FQCL -> packagename and clasname
@
hashcode -> hexdecimal value
generated for ever single
object

Hashtable -> we will study the
hashcode

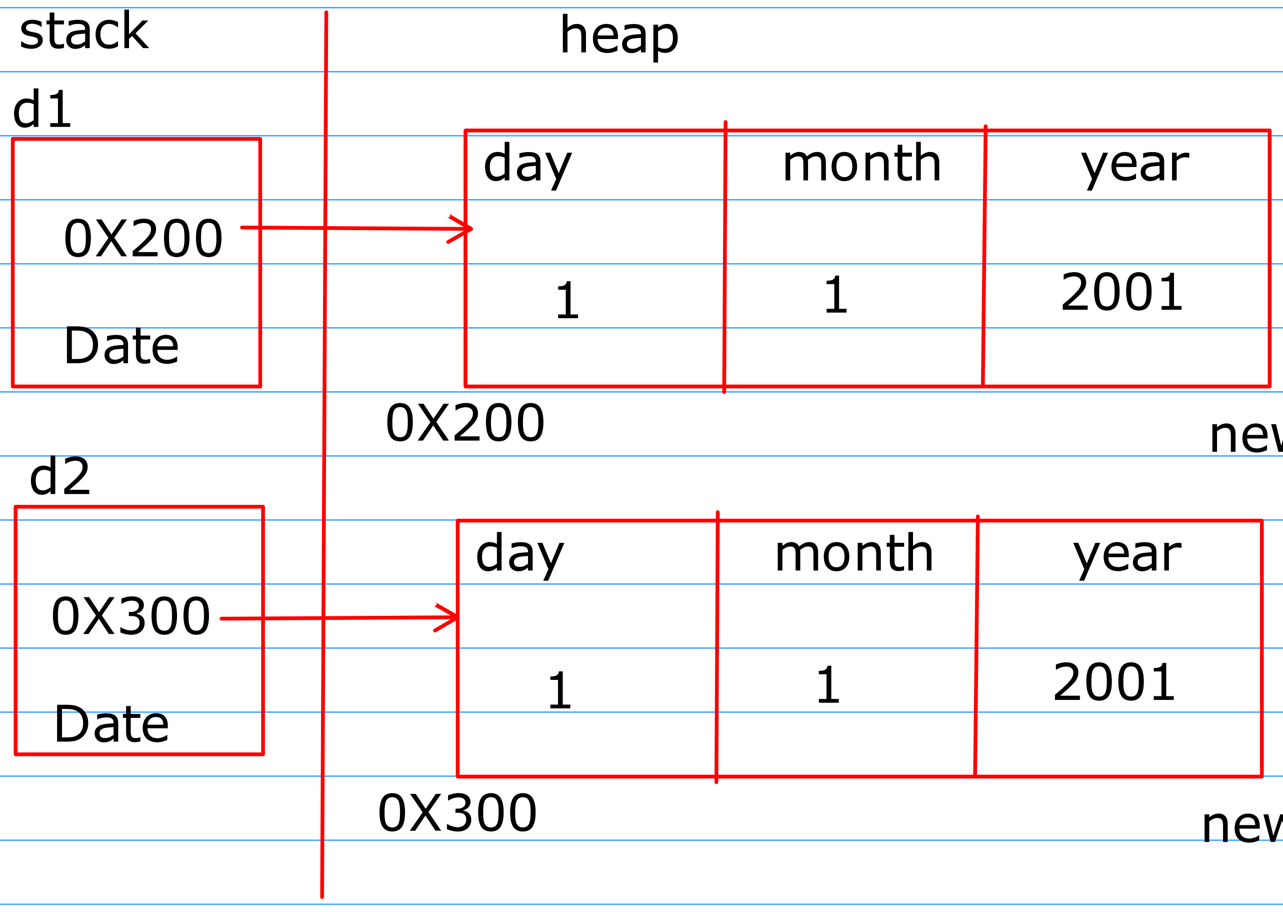


Date d1 = new Date(1,1,2001);

new Date(1,1,2001)

Date d2 = d1;

d1==d2
== checks for the references



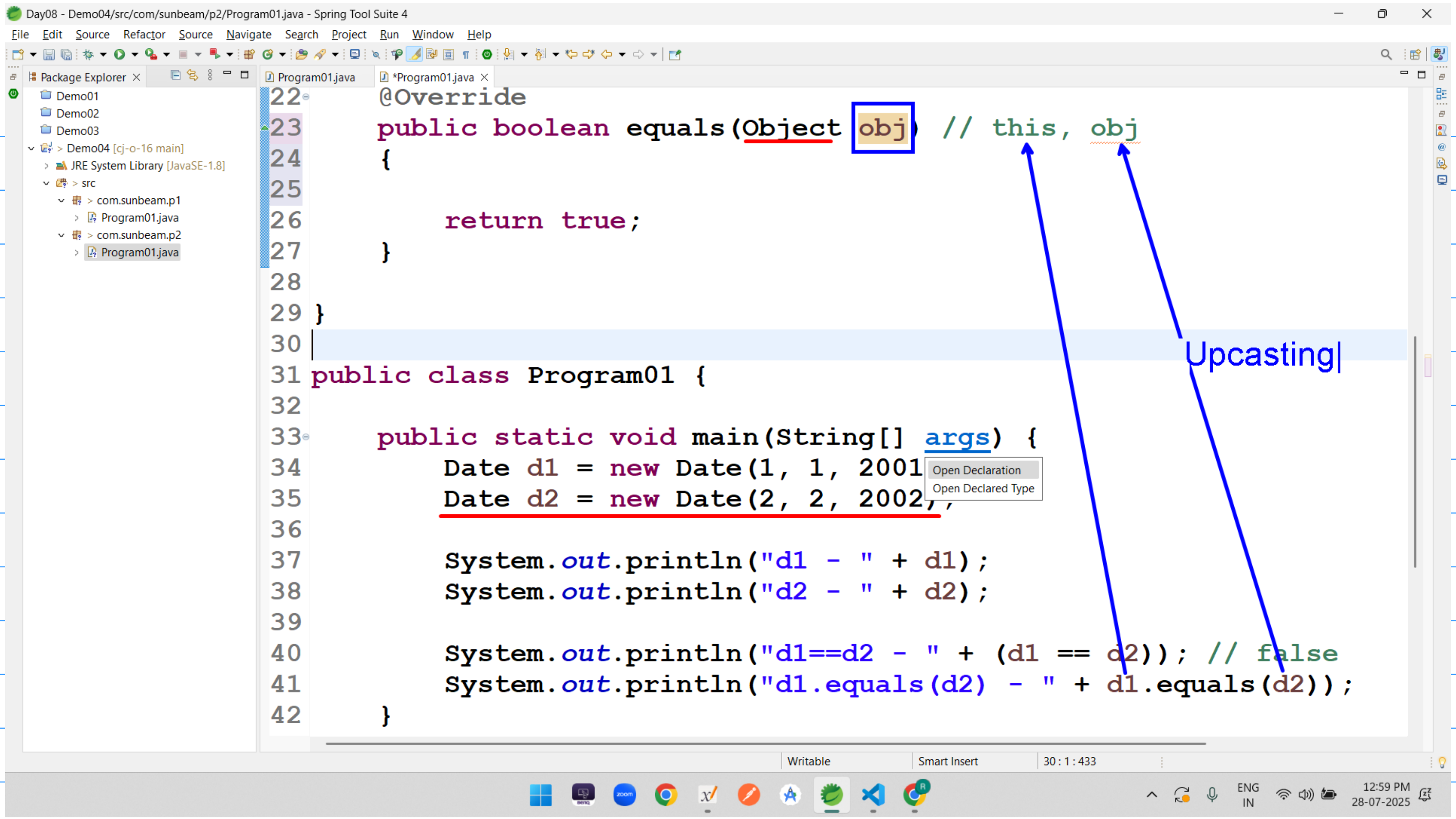
Date d1 = new Date(1,1,2001);

new Date(1,1,2001)

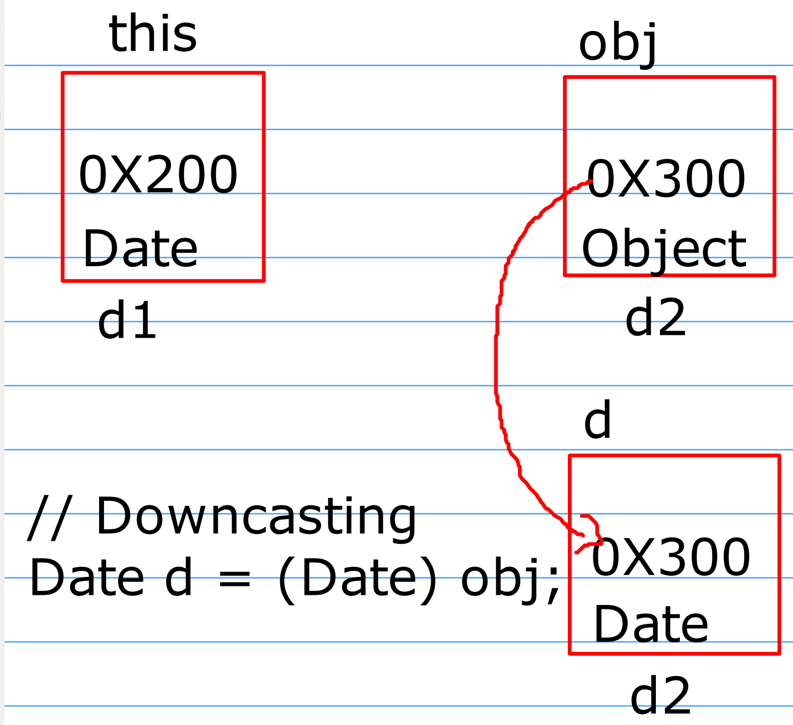
Date d2 = new Date(1,1,2001);

new Date(1,1,2001)

d1==d2
== checks for the references



```
    @Override
    public boolean equals(Object obj) // this, obj
    {

        return true;
    }

}

public class Program01 {

    public static void main(String[] args) {
        Date d1 = new Date(1, 1, 2001
        Date d2 = new Date(2, 2, 2002);

        System.out.println("d1 - " + d1);
        System.out.println("d2 - " + d2);

        System.out.println("d1==d2 - " + (d1 == d2)); // false
        System.out.println("d1.equals(d2) - " + d1.equals(d2));
    }
```



// Downcasting
Date d = (Date) obj;

Java 7