

# Constructor & Inheritance

## Pertemuan 6



# Topics

---

1. **Constructor**
2. **Inheritance**
3. **Function Overloading & Overriding**



# 1. Constructor

---

- Constructor adalah method yang secara otomatis dipanggil/dieksekusi saat sebuah class diinstansiasi.
- Nama constructor harus sama dengan nama class-nya.
- Sama halnya dengan method, constructor juga dapat memiliki satu atau lebih dari satu parameter.



# Contoh implementasi constructor

---

```
public class Mahasiswa
{
    private String nim, nama;
    public Mahasiswa()
    {
        this.nim = "";
        this.nama = "";
    }
}
```



# Contd..

---

```
public class Mahasiswa
{
    private String nim, nama;
    public Mahasiswa()
    {
        this.nim="";
        this.nama = "";
    }
    public Mahasiswa(String nim, String nama)
    {
        this.nim = nim;
        this.nama = nama;
    }
}
```

Contoh class yang memiliki lebih dari satu constructor disebut **multiple constructor**



# Function Overloading

---

- Function **Overloading** merupakan suatu kondisi dimana suatu class memiliki fungsi yang sama tetapi deklarasi dan parameternya berbeda.



# Contoh implementasi :

---

```
public class Matematika
{
    ....
    public int Tambah (int a, int b)
    {
        return a + b;
    }
    public int Tambah (int a, int b, int c)
    {
        return a + b + c;
    }
}
```



## 2. Inheritance

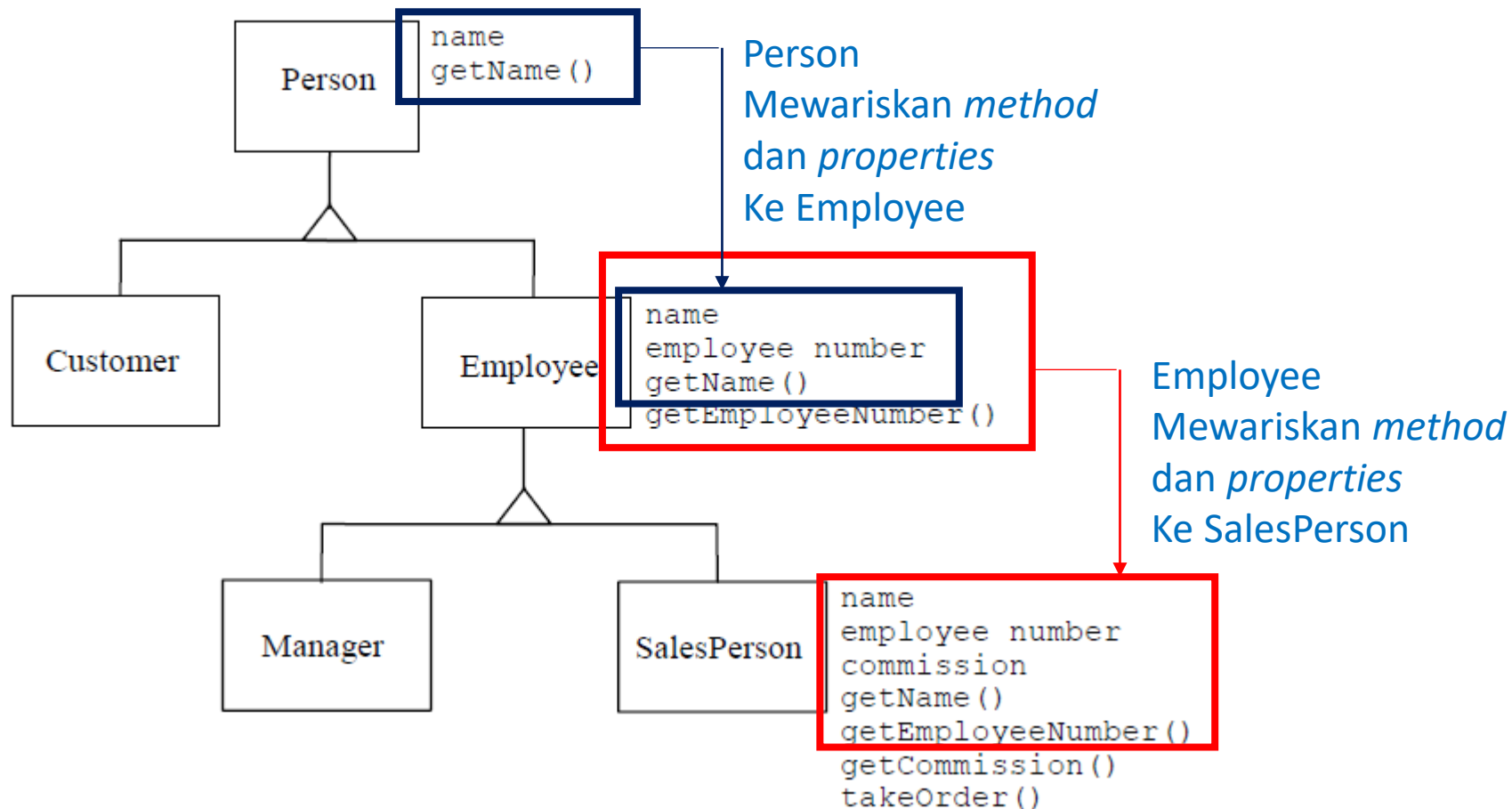
---

- **Inheritance** secara definisi merupakan suatu proses untuk mewariskan suatu sifat (baik method maupun properties) dari superclass ke subclass.
- Inheritance dapat meningkatkan “software reuse”
- Inheritance dapat pula dikatakan sebagai kemampuan dari subclass untuk mengambil rantai pewarisan dari superclassnya.





# Contoh hirarki inheritance :



# Contd..

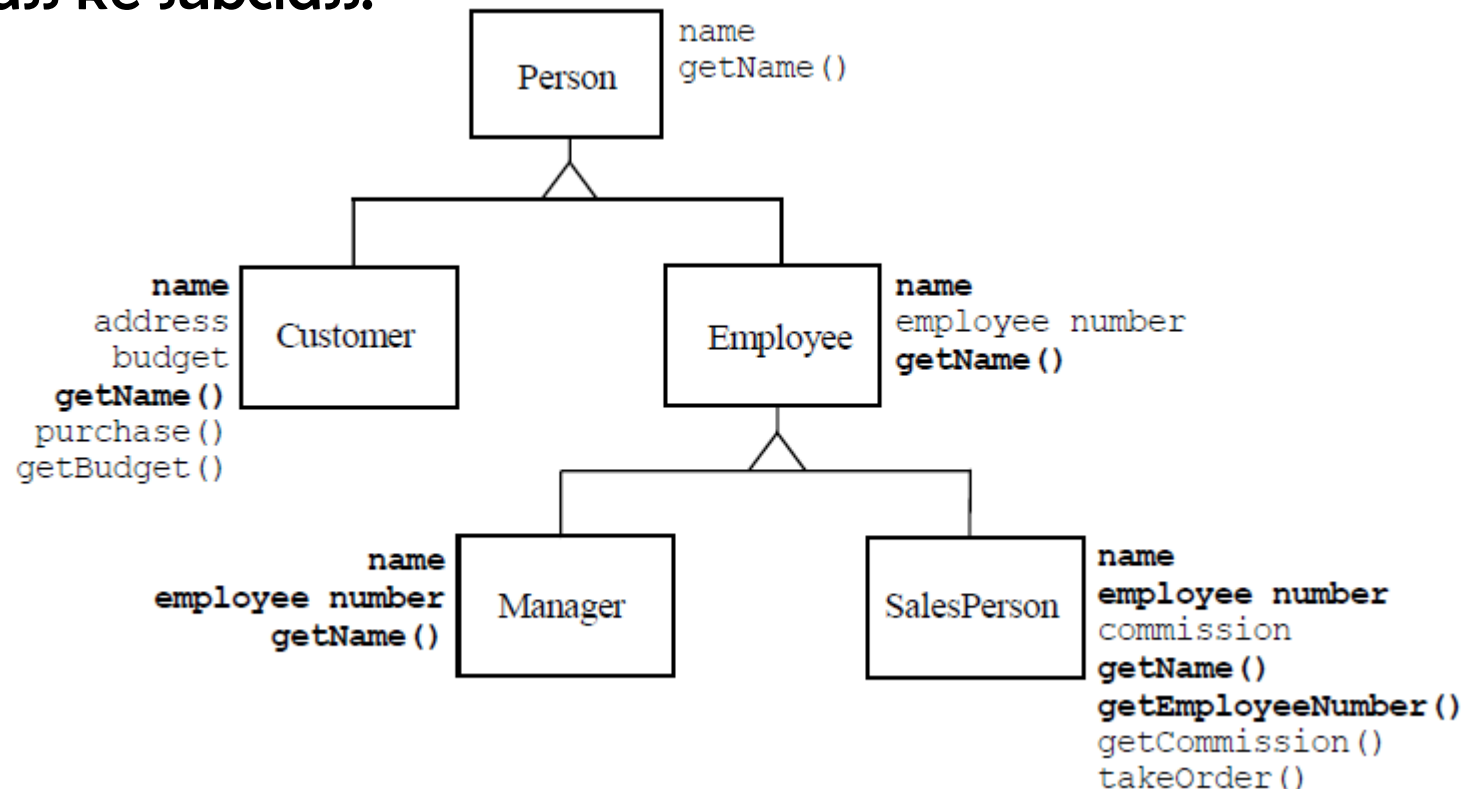
---

- Berdasarkan gambar inheritance tersebut, **SalesPerson** dapat didefinisikan sebagai kombinasi dari :
  - Kelas Employee
  - Kelas Person
  - Serta atribut dan method yang didefinisikan sendiri



# Classes with Inherited Properties

- Perhatikan hirarki method dan attribute yang diwariskan superclass ke subclass.



# Implementasi inheritance :

---

```
Class Person {  
    Attributes :  
        name  
    Methods :  
        getName()    {return name}  
}
```

```
class Employee extends Person {  
    ...  
}
```

```
class Manager extends Employee {  
    ...  
}
```

```
class SalesPerson extends Employee {  
    ...  
}
```



## Contd..

---

- Implementasi inheritance dilakukan dengan menggunakan statement “extends”.  
Struktur kodenya adalah sebagai berikut :

```
namaSubclass extends namaSuperclass  
{  
    .... //definisi  
}
```



# Contd..

---

- Statement “**super**” digunakan oleh subclass untuk memanggil **constructor** atau **method** yang ada pada superclass-nya.
- Contoh memanggil constructor pada superclass-nya :
  - **super()**
  - **super(parameter)**
- Contoh memanggil method pada superclass-nya :
  - **super>NamaMethod(parameter)**



# Function overriding

---

- Merupakan suatu kondisi dimana method pada subclass mereplace / override method hasil inheritance dari superclassnya.



# Contoh :

---

```
public class food
{
    public void eat()
    {
        System.out.println("i'am the eat method");
    }
}
```

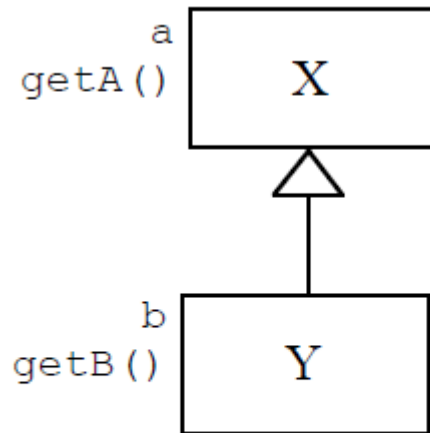
```
public class tuna extend food
{
    public void eat()
    {
        System.out.println("i'am the new eat method");
    }
}
```



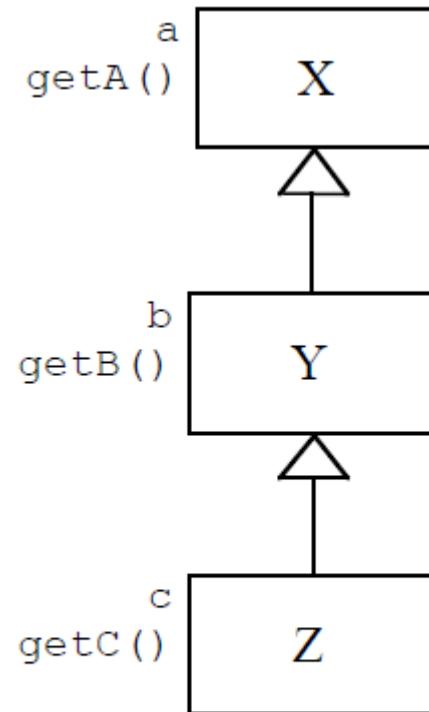


# Inheritance Chain

---



Single-level Single  
Inheritance Chain

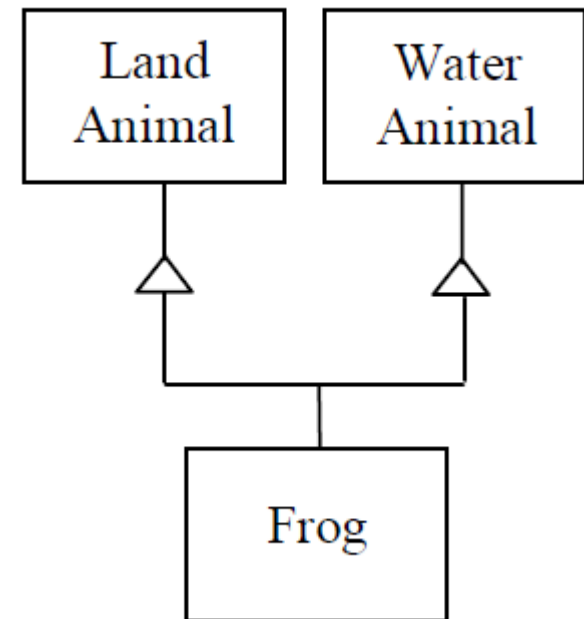


Multiple-level Single  
Inheritance Chain

# Multiple Inheritance

---

- Inheritance yang dilakukan pada lebih dari satu superclass disebut multiple inheritance.
- Kendala : extend hanya berlaku pada satu superclass.
- Solusi : interface



# Final Classes

---

- Agar membuat tidak dapat di-inherit, maka perlu dibuat dengan “final”
- Kita dapat menuliskannya sebagai modifier
- Contoh :
  - Public **final** class Executive extends Manager {..}



# Demo inheritance

---

- Demo di netbeans



# References :

---

1. **Object oriented programming and Java  
2<sup>nd</sup> Edition – Chapter 6**

