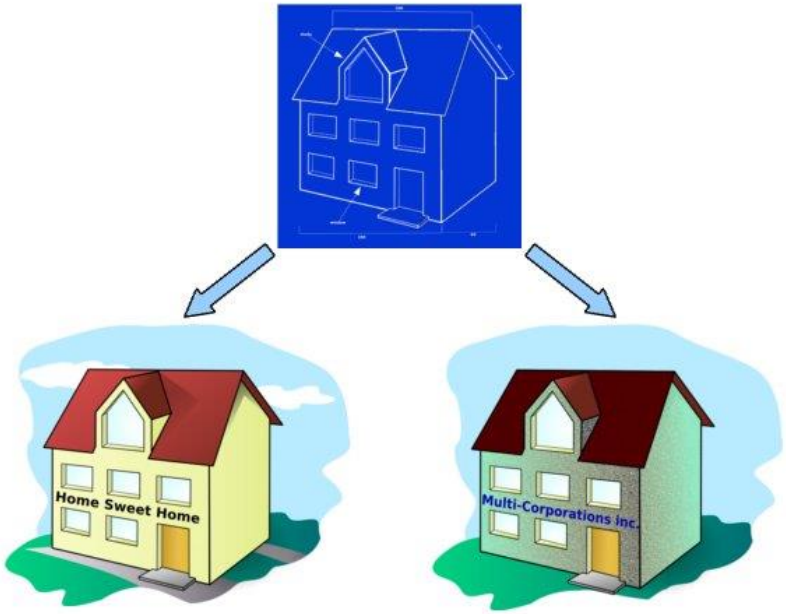
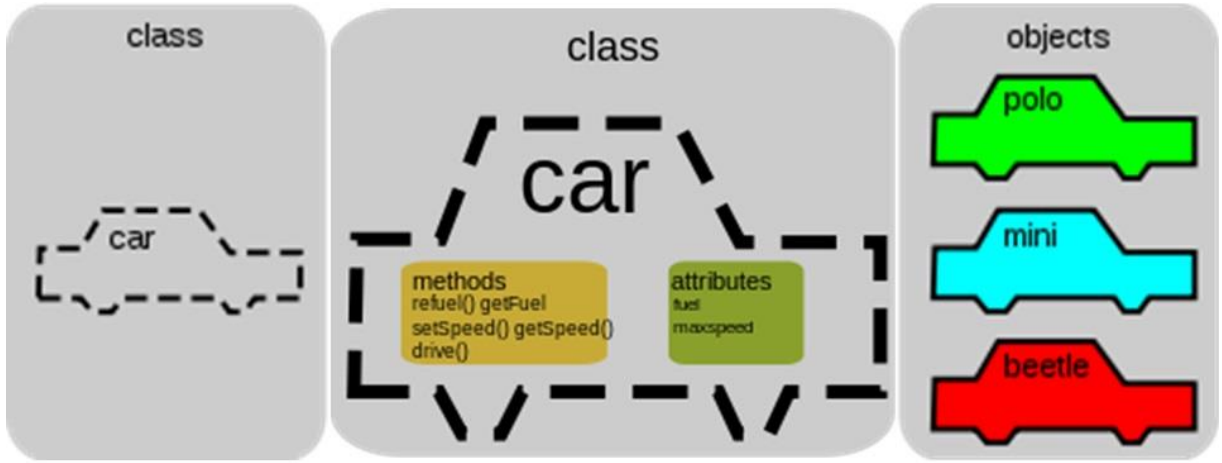
# Java

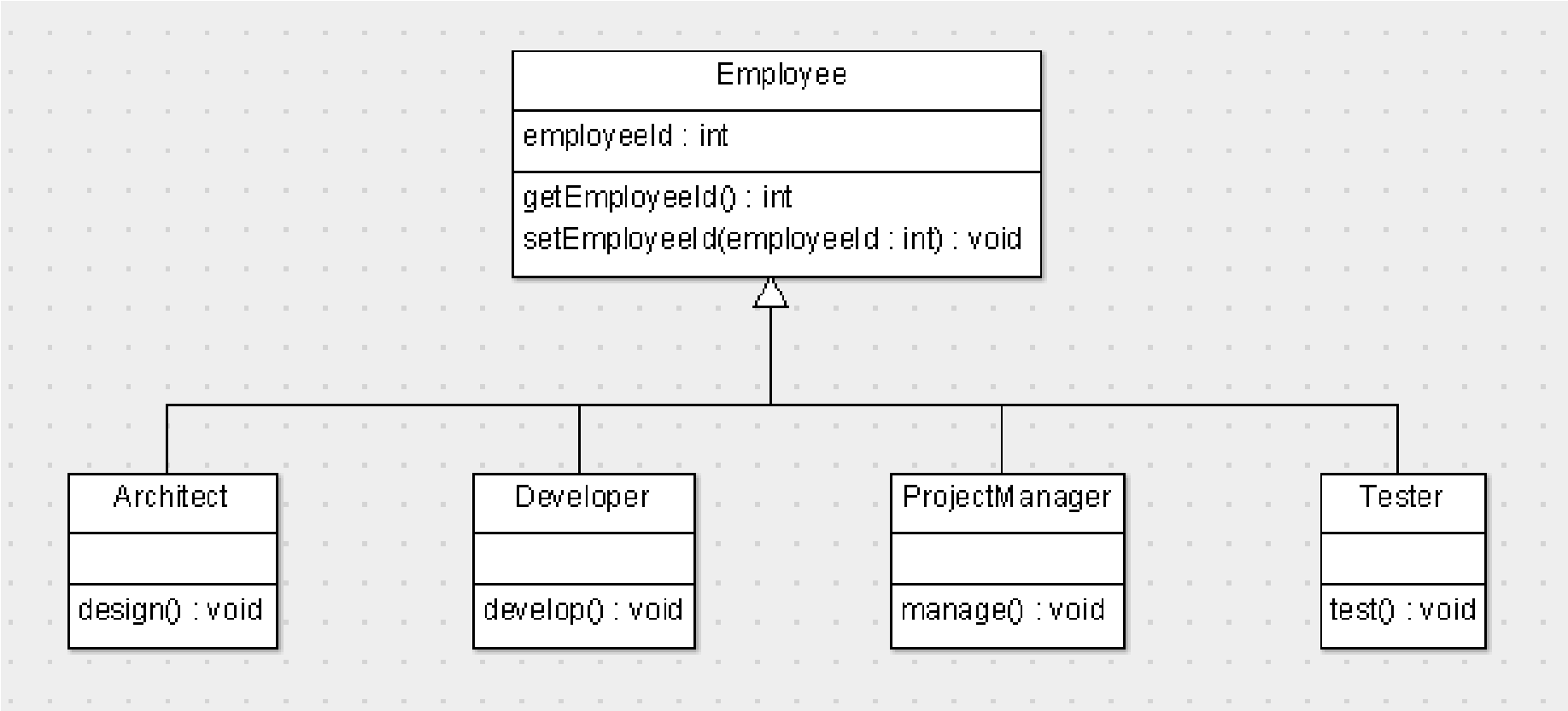
Sridhar A

# Class and Objects



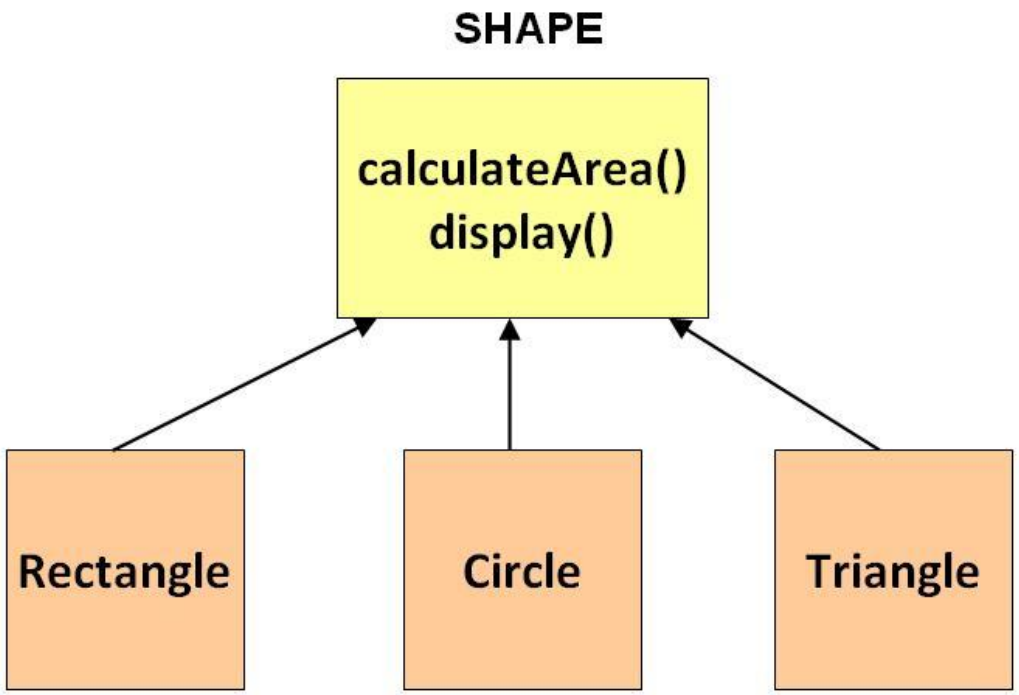



# Inheritance



# Abstract classes and methods

- A class which contains the **abstract** keyword in its declaration is known as abstract class.
- Abstract classes may or may not contain *abstract methods*, i.e., methods without body ( public void get(); )
- But, if a class has at least one abstract method, then the class **must** be declared abstract.
- If a class is declared abstract, it cannot be instantiated.
- To use an abstract class, you have to inherit it from another class, provide implementations to the abstract methods in it.
- If you inherit an abstract class, you have to provide implementations to all the abstract methods in it.



# Super keyword

```
public class Object{
    public Object() {
        ...
    }
}
```

```
public class ClassA {
    public ClassA() {
        super();
        System.out.println("Class A constructor");
    }
}
```

```
public class ClassB extends ClassA{
    public ClassB() {
        super();
        System.out.println("Class B constructor");
    }
}
```

# this keyword

```
public class MyNumber {

    int x = 3;

    public void setX() {

        int x = 4;

        System.out.printn(x);

        System.out.printn(this.x);
    }
}
```

```
class Account{
int a;
int b;
public void setData(int a , int b){
    this. a=a;
    this. b=b;
}
public static void main(string args[]){
Account obj = new Account();

}
```

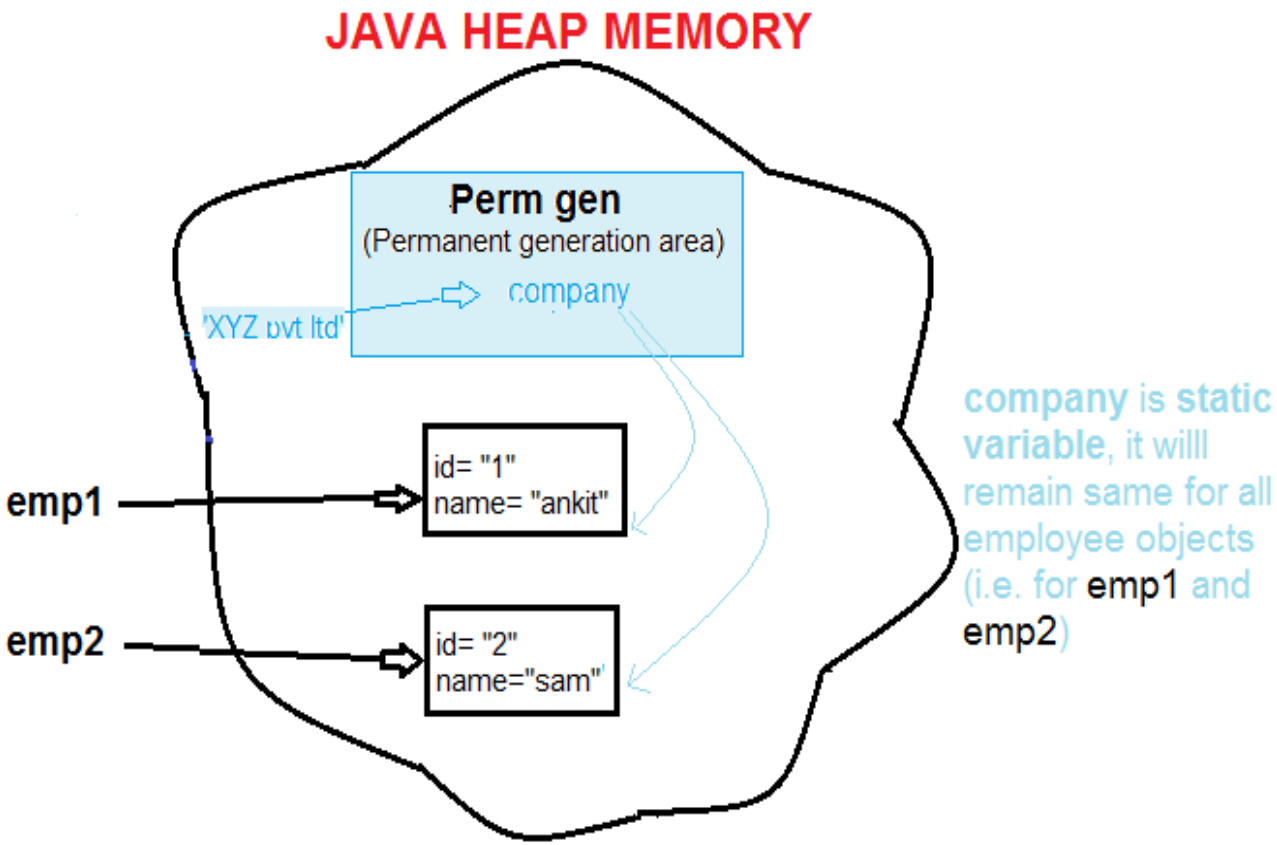
use keyword "This" to differentiate instance variable from local variable

# Static keyword

The **static keyword** in java is used for memory management mainly.

The static can be:

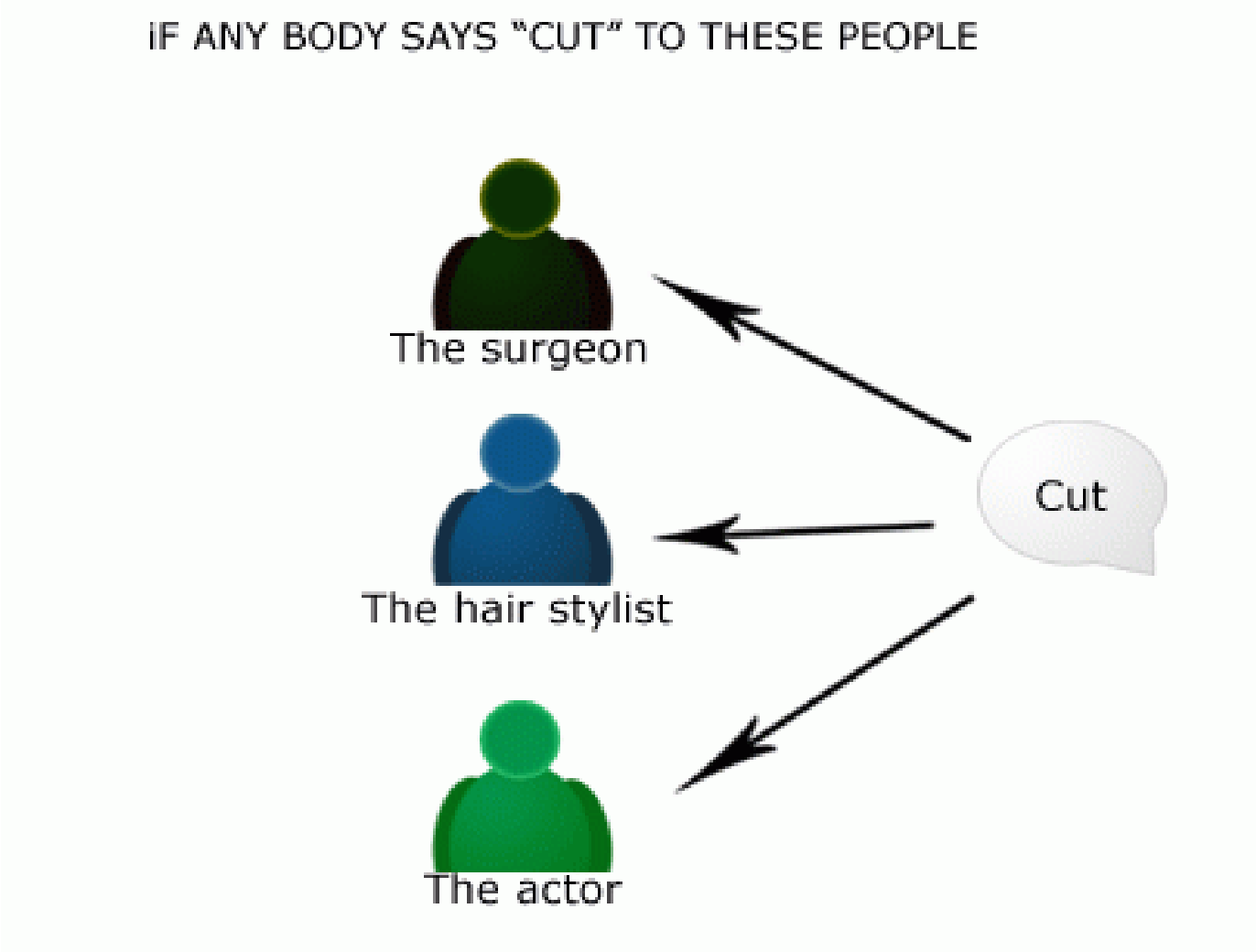
- variable / class variable
- method / class method
- block
- nested class



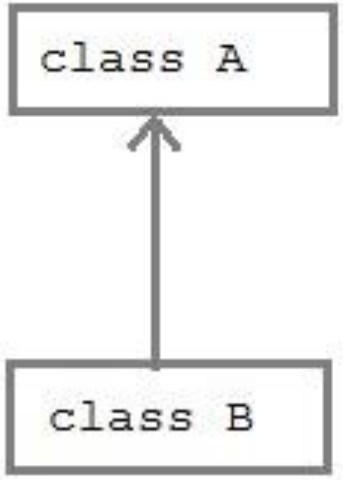
# Final keyword

- Restrict changing value of a variable
- Restrict method overriding
- Restrict Inheritance

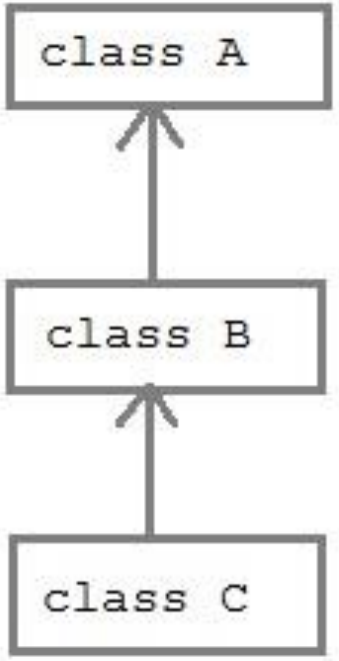
# Polymorphism



# Types of Interitance

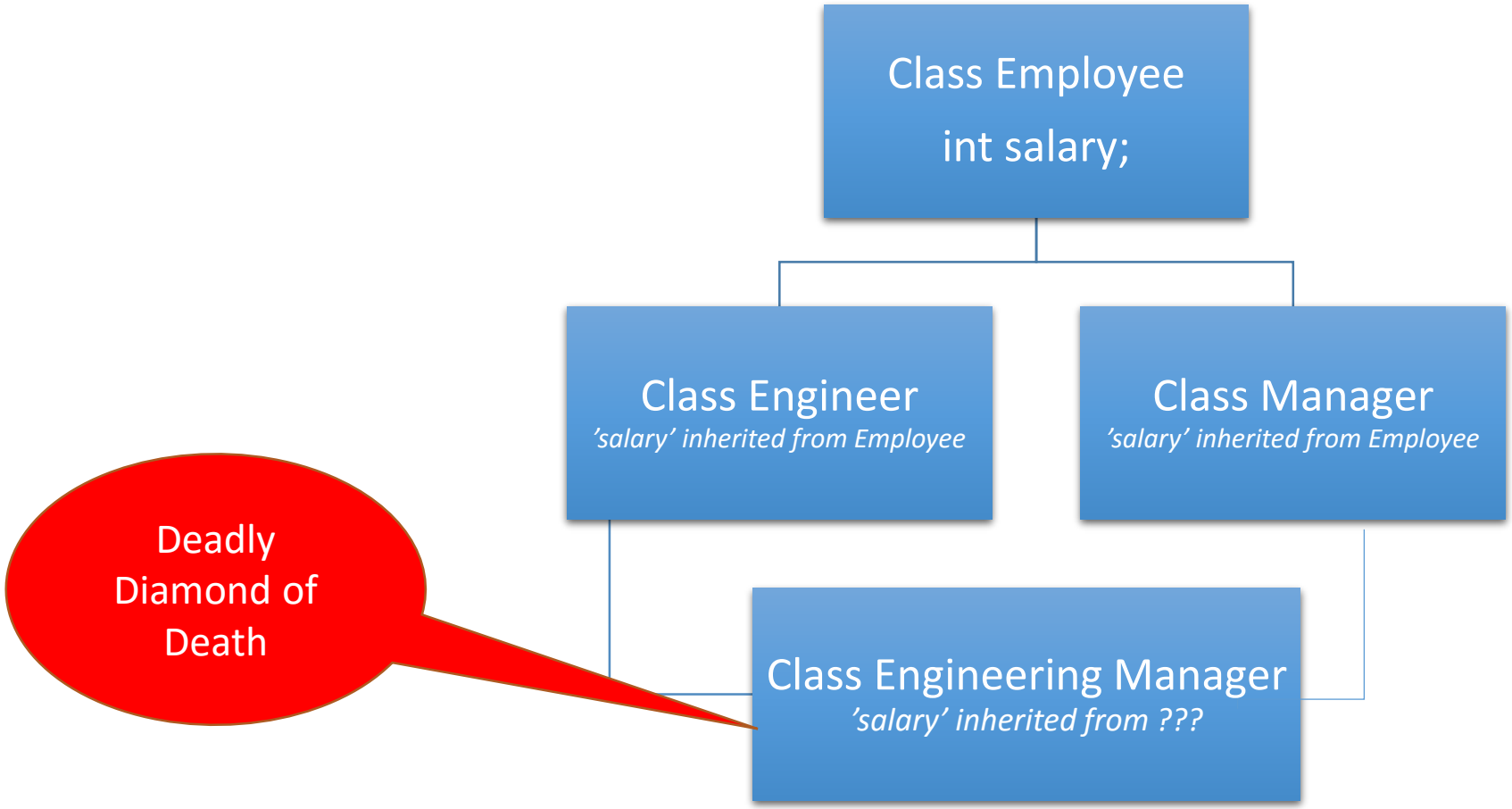



**Simple Inheritance**



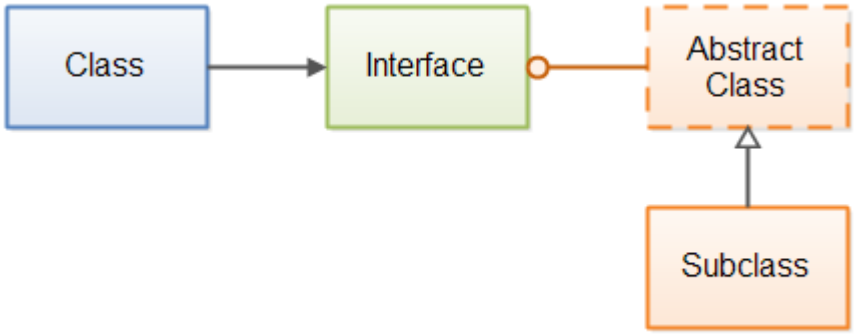

**Multilevel inheritance**

# Multiple Inheritance



# Interfaces

```
import java.lang.*;
//multiple import statements can be used here
public interface IntrfaceName
{
//Any number of final, static fields
//Any number of abstract method declarations
}
```



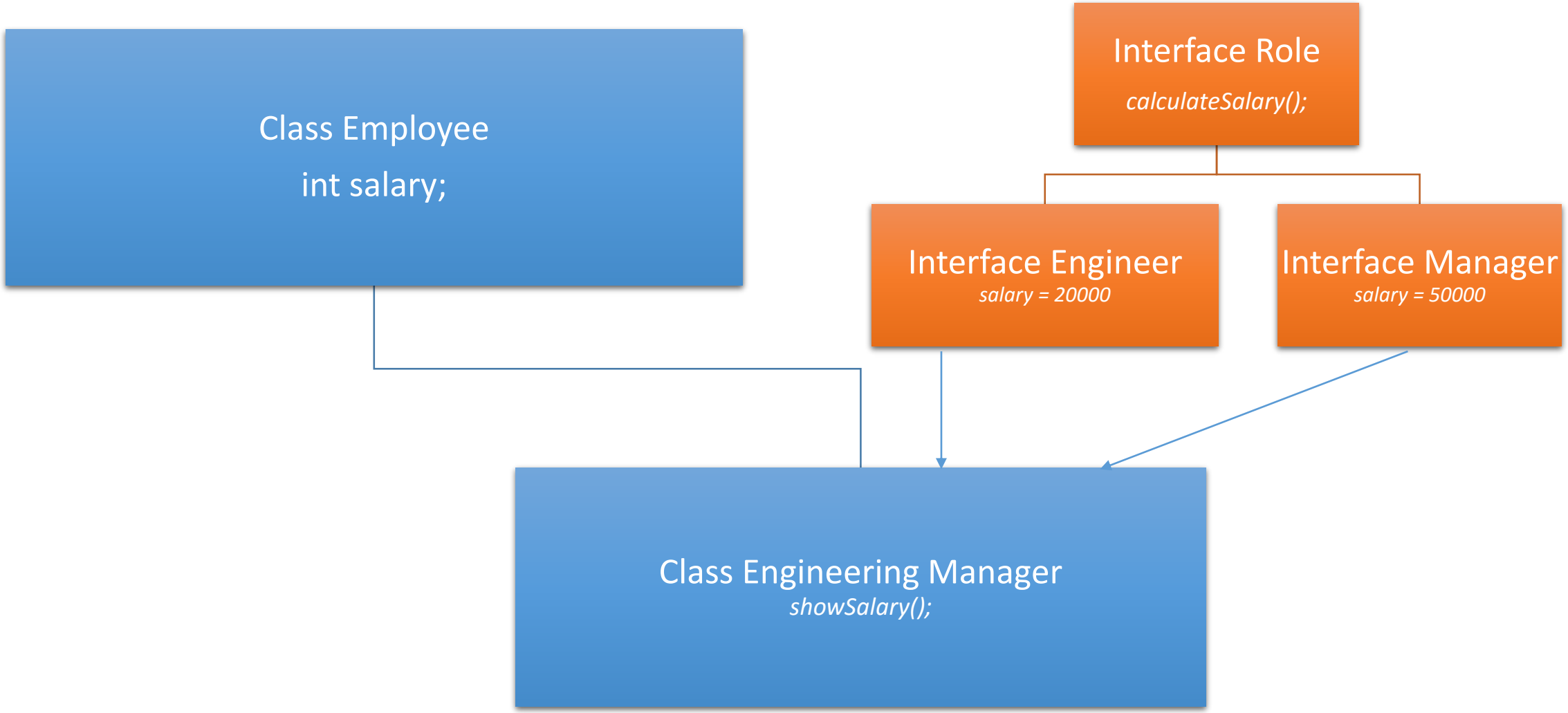
# Interfaces

- Used for achieving full data abstraction
- Each and every method in interface is also implicitly abstract
- All methods of an interface are implicitly public
- Use “implements” keyword in class
- All the classes implementing interfaces, must override the methods declared in the interfaces implemented.

```
// One interface can extend another.
interface A {
  void meth1();
  void meth2();
}

// B now includes meth1() and meth2() - it adds meth3().
interface B extends A {
  void meth3();
}

// This class must implement all of A and B
class MyClass implements B {
  public void meth1() {
    System.out.println("Implement meth1().");
  }
  public void meth2() {
    System.out.println("Implement meth2().");
  }

  public void meth3() {
    System.out.println("Implement meth3().");
  }
}

class IFExtend {
  public static void main(String args[]) {
    MyClass ob = new MyClass();

    ob.meth1();
    ob.meth2();
    ob.meth3();
  }
}
```

B inherits A.

# Multiple Inheritance using Interfaces



# Constructors

- Interface cannot have constructor.
- Constructors cannot be private.
- A constructor cannot be abstract, static, final, native or synchronized
- A constructor can be overloaded.
- Constructors cannot return a value.
- Constructors do not have a return type; not even void.
- Abstract class can have constructor.
- Constructors name must be similar to that of class name inside which it resides.
- Constructors are automatically called when an object is created.

# Quiz

Method with the same name or different return type and difference in the parameters either in number or type is known as

A. Function overloading

B. Compile Time Overloading

# Quiz

Which of the following is not a part of OOP?

A. Type checking

B. Multitasking

C. Polymorphism

D. Information hiding

# Quiz

Which of the following is not a part of OOP?

A. Type checking

B. Multitasking

C. Polymorphism

D. Information hiding

# Quiz

To call a base class constructor in a derived class, is it needed to call the base class initializer.

A. True

B. False

# Quiz

Can objects of abstract classes be instantiated

A. True
B. False

# Quiz

Mention two forms of polymorphism

# Quiz

The process by which one object can acquire the properties of another object

A. Encapsulation

B. Inheritance

C. Polymorphism

# Quiz

Constructors are used to

A. To build a user interface.

B. Free memory.

C. Initialize a newly created object.

D. To create a sub class

# Quiz

An object that has more than one form is referred to as

A. Inheritance

B. Interface

C. Abstract class

D. Polymorphism

# Quiz

- Information Hiding can also be termed as

A. Data hiding

B. Encapsulation

C. Inheritance

# Quiz

When deriving from a private base class, the public, protected and private members of the base class become private members of the derived class.

A. True

B. False

# Quiz

Pick the term that relates to polymorphism

A. Dynamic binding

B. Dynamic allocation

C. Static typing

D. Static allocation

# Quiz

Two or more methods with same name in the same class with different arguments is called as

A. Method overriding

B. Method overloading

# Quiz

- Main method can be overriden

A. True

B. False

# Quiz

Keyword which is used to access the method or member variables from the base class

A. super

B. using

C. this

D. final

# Quiz

An _____ cannot provide any code at all, can provide only the signature.

A. Abstract class

B. Interface

# Quiz

When sub class declares a method that has the same type arguments as a method declared by one of its superclass, it is termed as

A. Method overriding

B. Method overloading

C. Operator overloading

D. Operator overriding

# Quiz

Static methods cannot be accessed directly from the class level.

A. True

B. False

# Quiz

- "What's the object-oriented way to become wealthy?"

# Thank You