# Core Java

UNIT 3 – Abstract Classes & Methods ,
Method Overriding

Single Inheritance

Multilevel Inheritance

Super

# Review

# Review – Identify the missing lines in the code if any

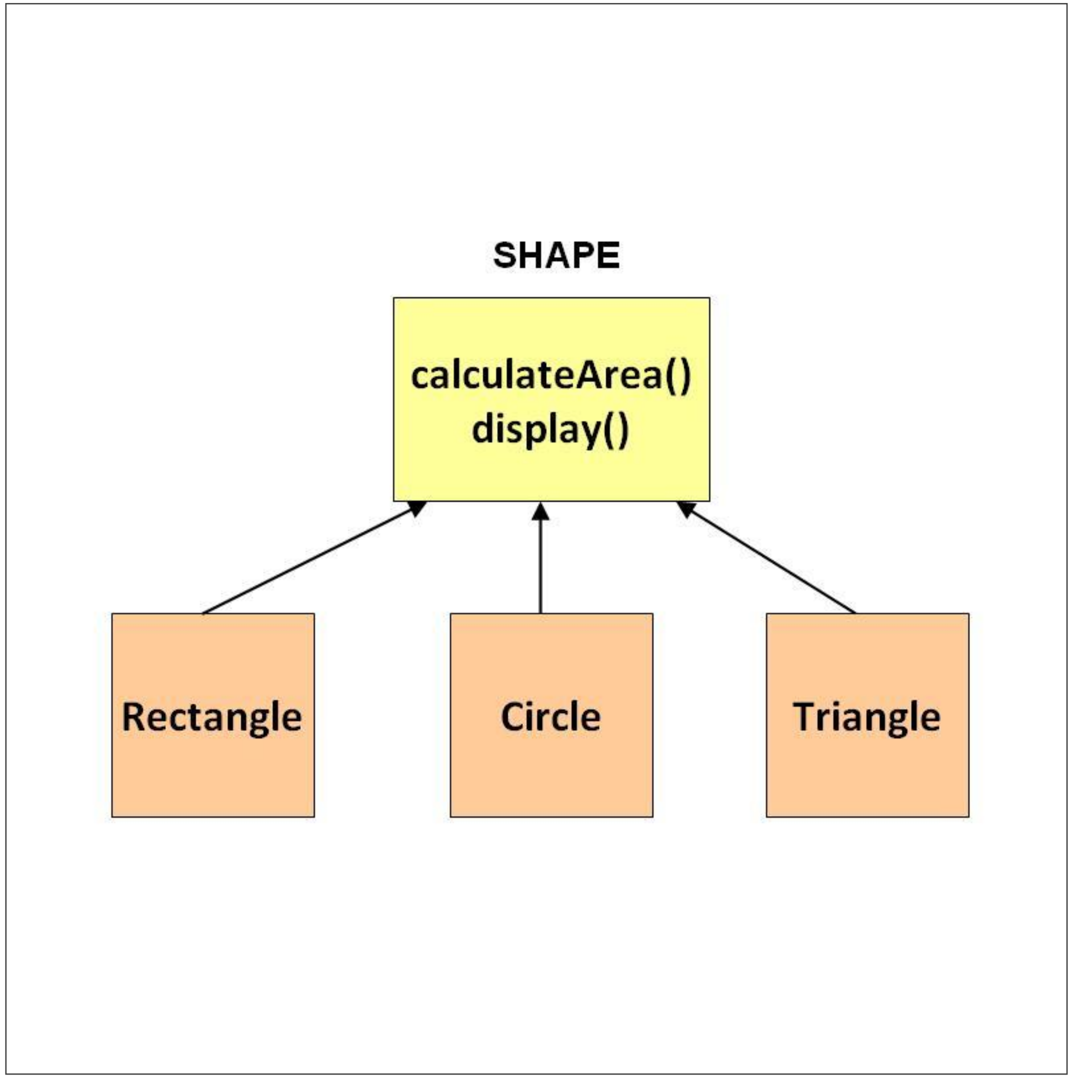
```
class Furniture{

        Furniture(){

        System.out.println("Furniture class Constructor");

                }

}

class Chair extends Furniture{

        Chair(){

        System.out.println("Chair class Constructor");

        }

}
```

```
class MainClass {
        public static void main(String args[])
        {
                Chair d=new Chair();
        }
}
```

# Review – Identify the missing lines in the code if any

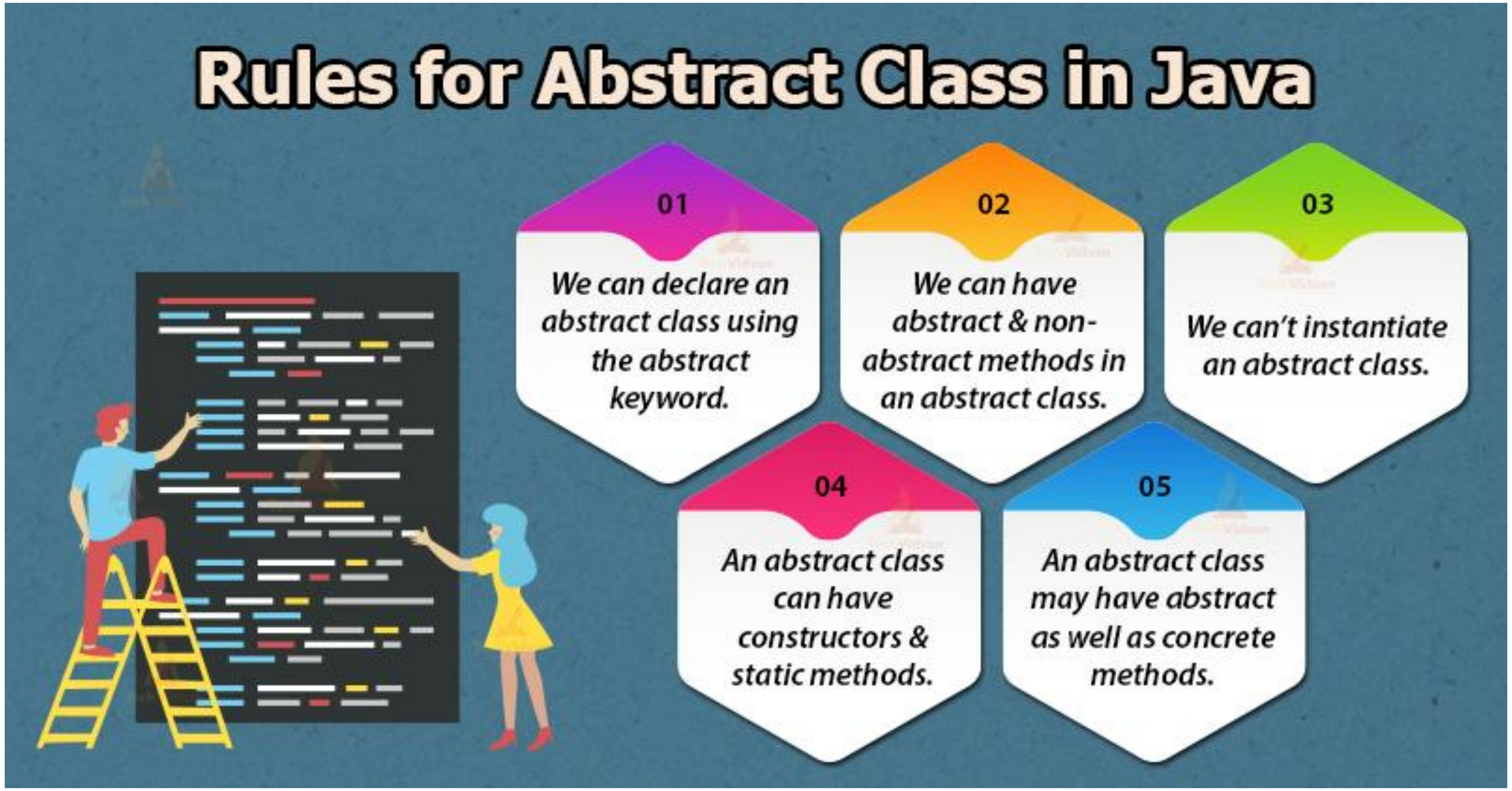
```
class Furniture{

        Furniture(String  type){

        System.out.println("Furniture class Constructor");

                }

}

class Chair extends Furniture{

        Chair(){  ——————→  super("softwood")

        System.out.println("Chair class Constructor");

        }

}
```

```
class MainClass {
        public static void main(String args[])
        {
                Chair d=new Chair();
        }
}
```



# Abstraction in Java

- **Abstraction** is a process of hiding the implementation details and showing only functionality to the user.



Rules for Abstract Class in Java

01 We can declare an abstract class using the abstract keyword.

02 We can have abstract & non-abstract methods in an abstract class.

03 We can't instantiate an abstract class.

04 An abstract class can have constructors & static methods.

05 An abstract class may have abstract as well as concrete methods.

# ABSTRACT CLASS

A class which is declared with the abstract keyword

# Abstract Methods

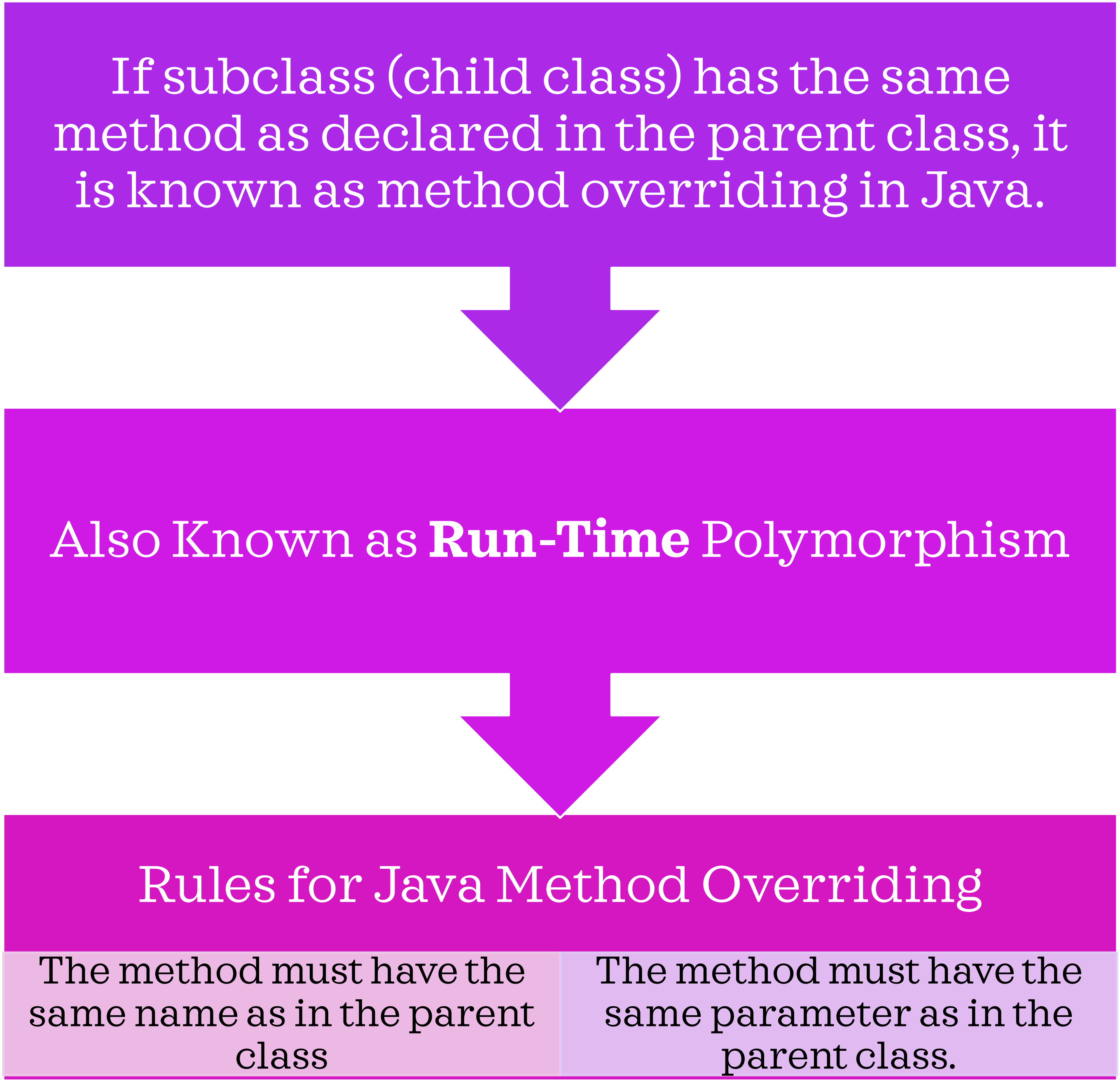
A method without body (no implementation) is known as abstract method.

A method must always be declared in an abstract class

Example:

```
public abstract int myMethod(int n1, int n2);
```

# Method Overriding



If subclass (child class) has the same method as declared in the parent class, it is known as method overriding in Java.

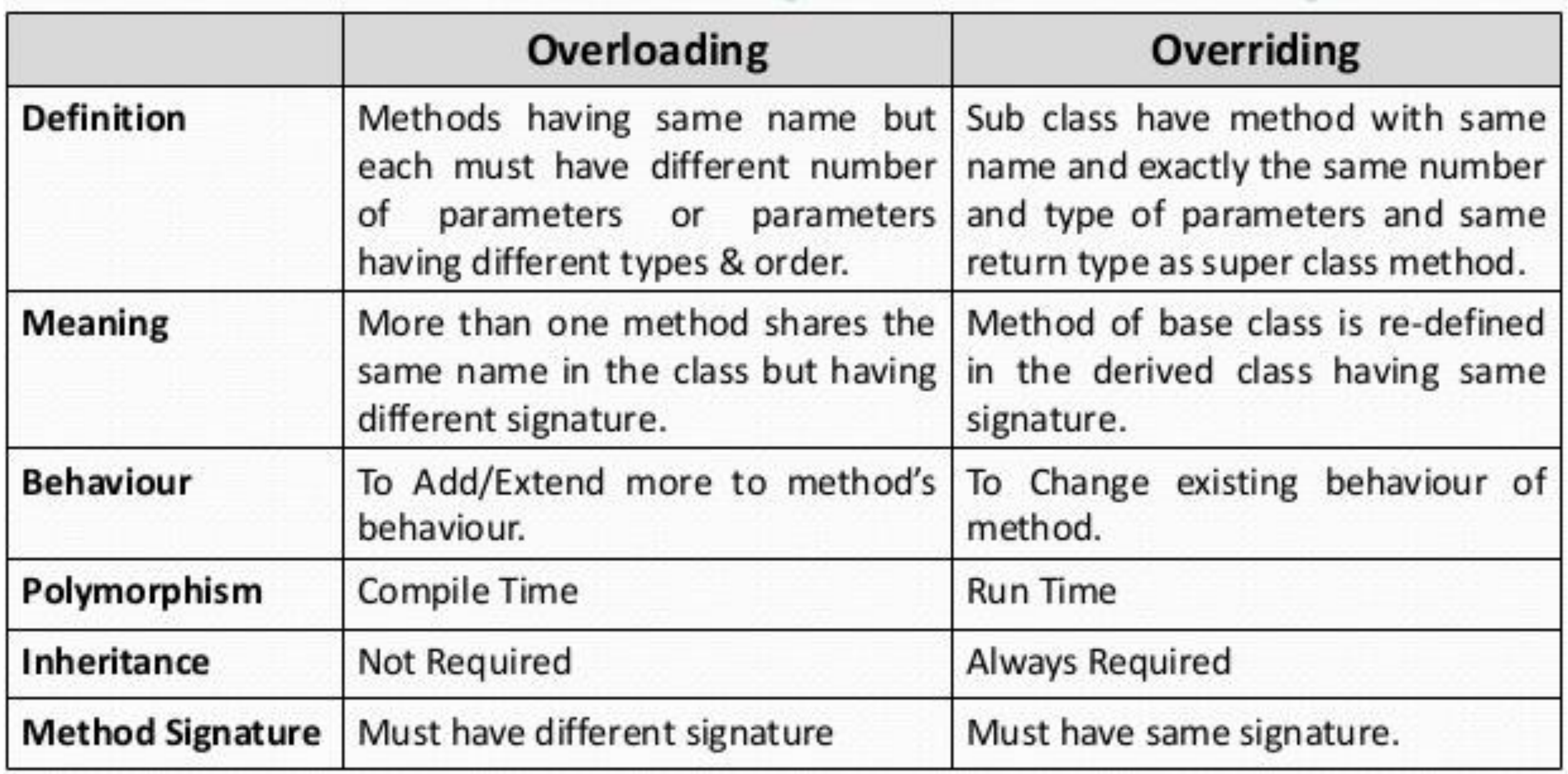
Also Known as **Run-Time** Polymorphism

Rules for Java Method Overriding

The method must have the same name as in the parent class

The method must have the same parameter as in the parent class.

# DISTINGUISH BETWEEN METHOD OVERLOADING AND OVERRIDING



| | Overloading | Overriding |
|---|---|---|
| **Definition** | Methods having same name but each must have different number of parameters or parameters having different types & order. | Sub class have method with same name and exactly the same number and type of parameters and same return type as super class method. |
| **Meaning** | More than one method shares the same name in the class but having different signature. | Method of base class is re-defined in the derived class having same signature. |
| **Behaviour** | To Add/Extend more to method's behaviour. | To Change existing behaviour of method. |
| **Polymorphism** | Compile Time | Run Time |
| **Inheritance** | Not Required | Always Required |
| **Method Signature** | Must have different signature | Must have same signature. |

# QUIZ

# THANK YOU