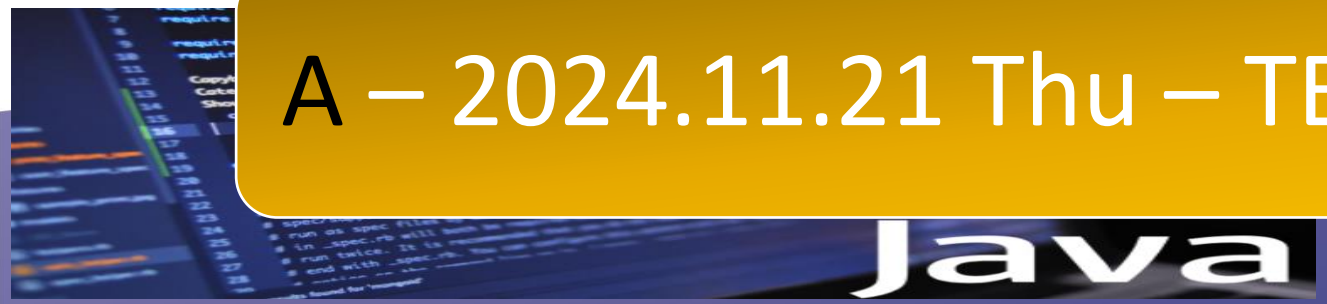
CJ Information

S – Unit 9 Inheritance

IC – Subclasses

HW - none

A – 2024.11.21 Thu – TEST (Unit4 & 5)



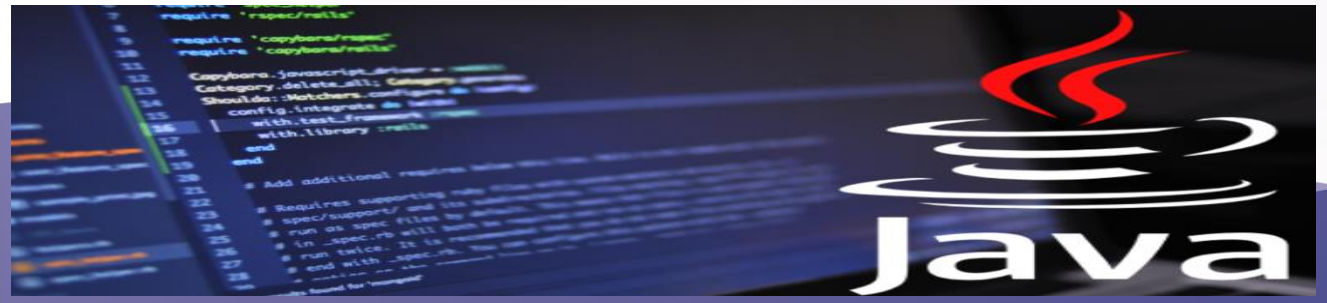
Inheritance in java

Outline :

- Inheritance Introduction
- Basic concepts behind Inheritance
- Types of Inheritance

What we learn today:

- Understand the meaning of Inheritance
- How to implement Inheritance in Java
- How Inheritance works in Java
- Examples

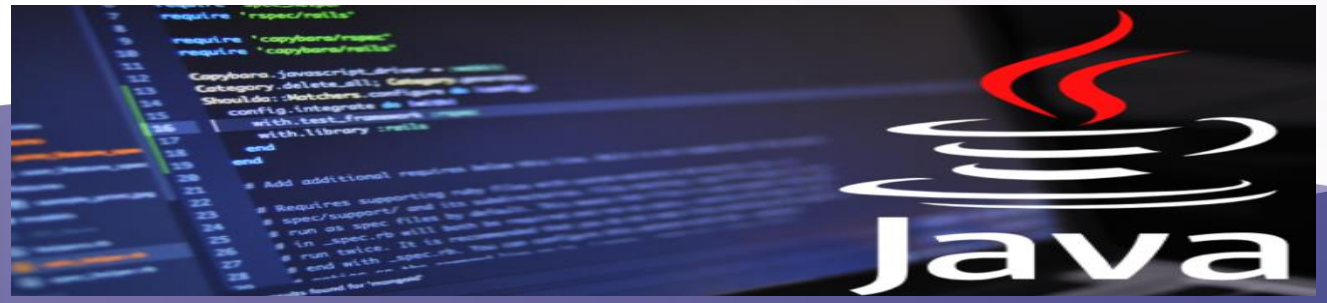


Inheritance in Java

Q : What is Inheritance?

A : *it is a mechanism by which one class is allowed to inherit the features (fields & methods) from another class*

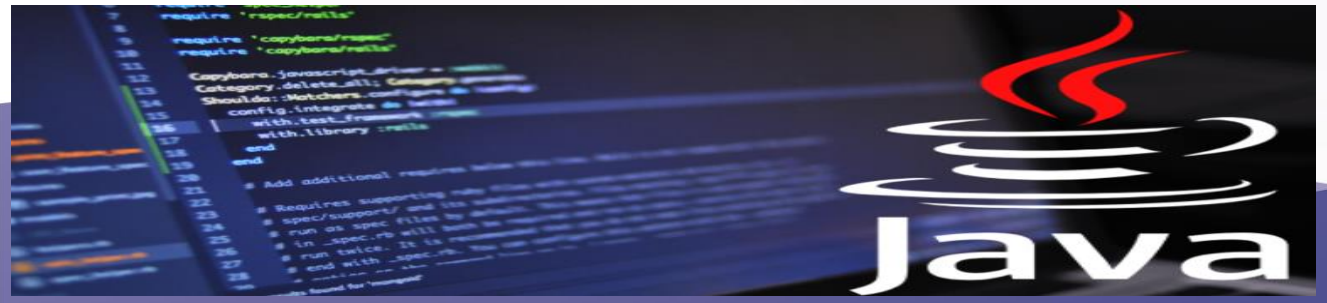
Based on the above definition, we can create a new class that can re-use the methods and the fields from another class.
In addition the new class can have new fields and methods specific for the class itself.



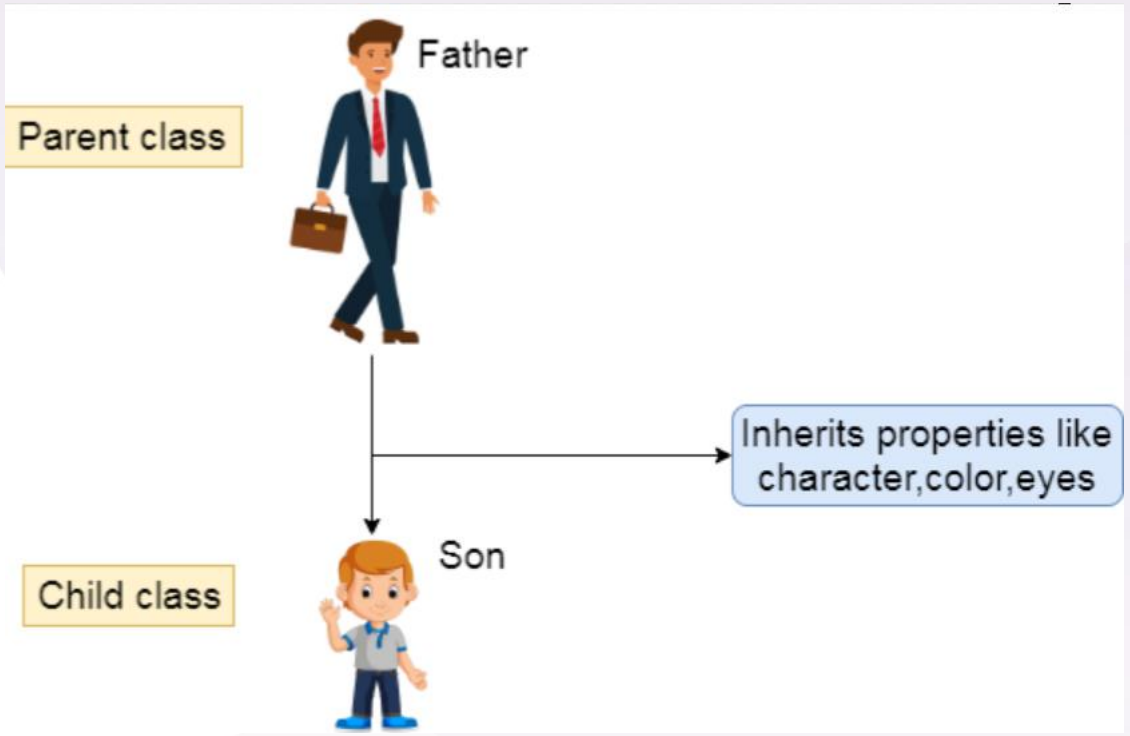
Inheritance in Java

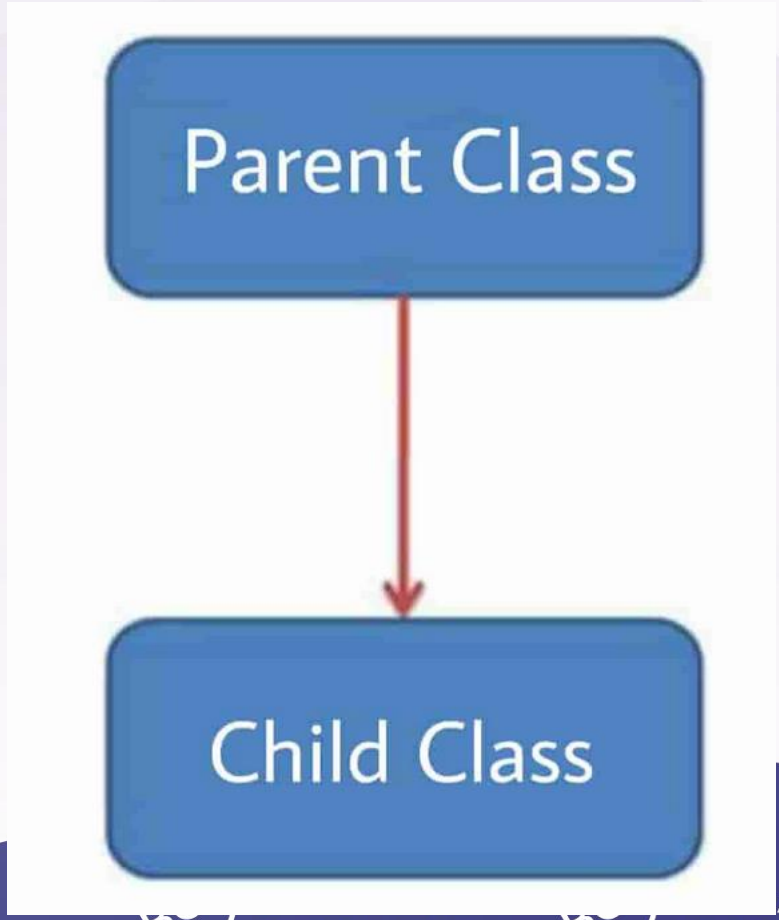
Why Do We Need Java Inheritance?

- **Code Reusability:** The code written in the Superclass is common to all subclasses. Child classes can directly use the parent class code.
- **Method Overriding:** Method Overriding is achievable only through Inheritance. It is one of the ways by which Java achieves Run Time Polymorphism.
- **Abstraction:** The concept of abstract where we do not have to provide all details is achieved through inheritance. Abstraction only shows the functionality to the user.



Inheritance in Java

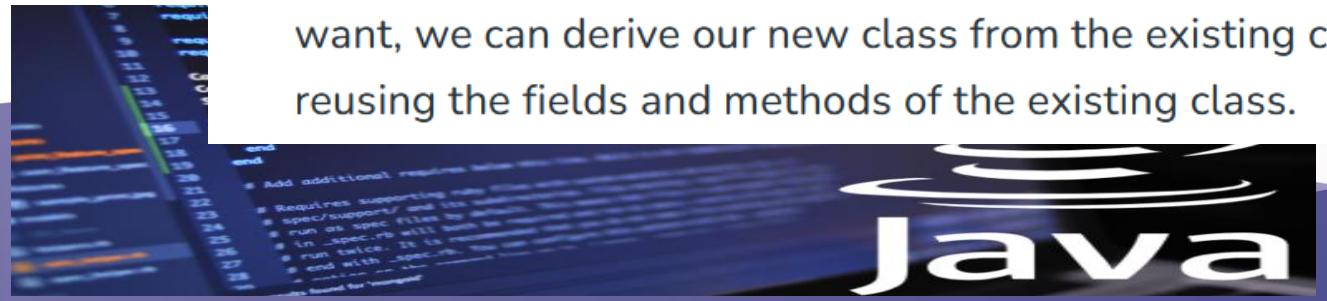




Inheritance in Java

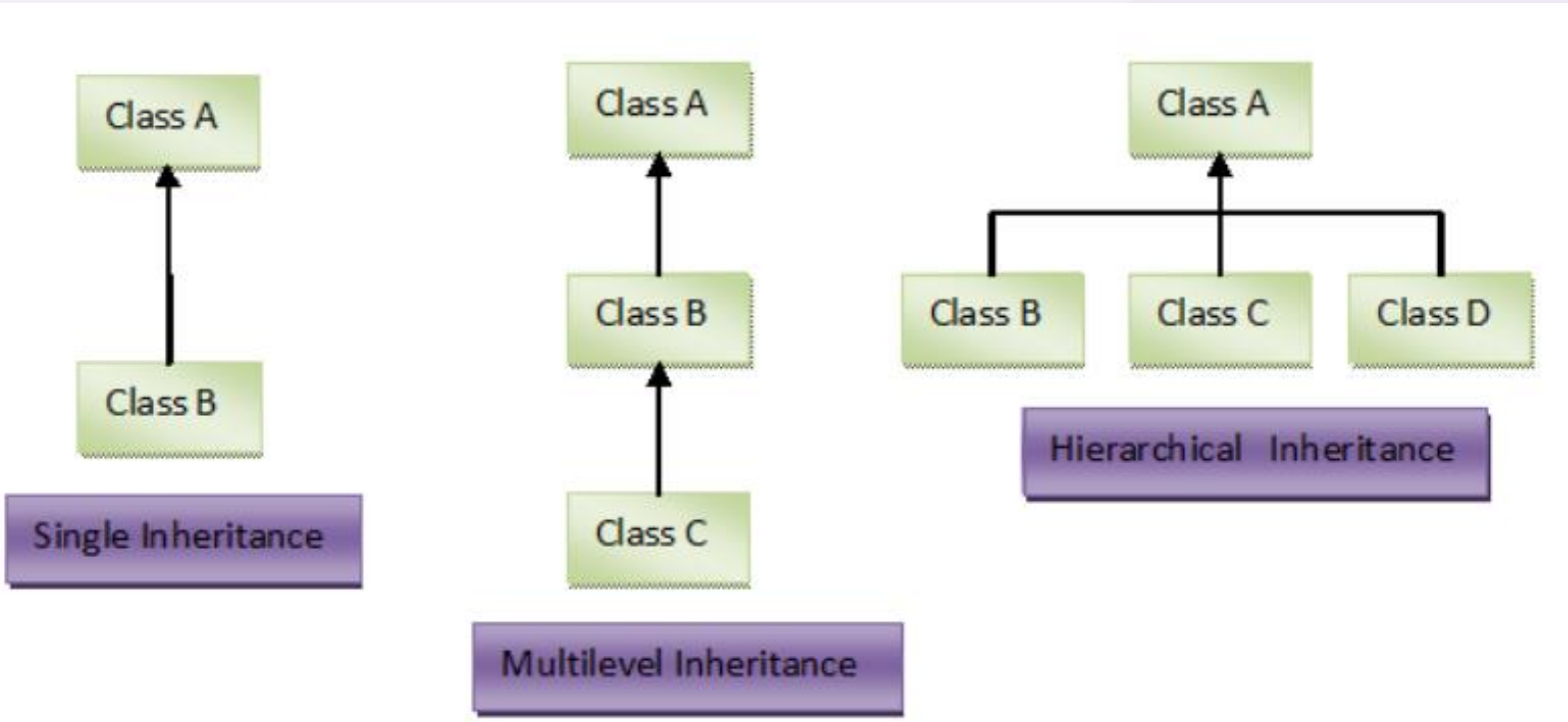
Important Terminologies Used in Java Inheritance

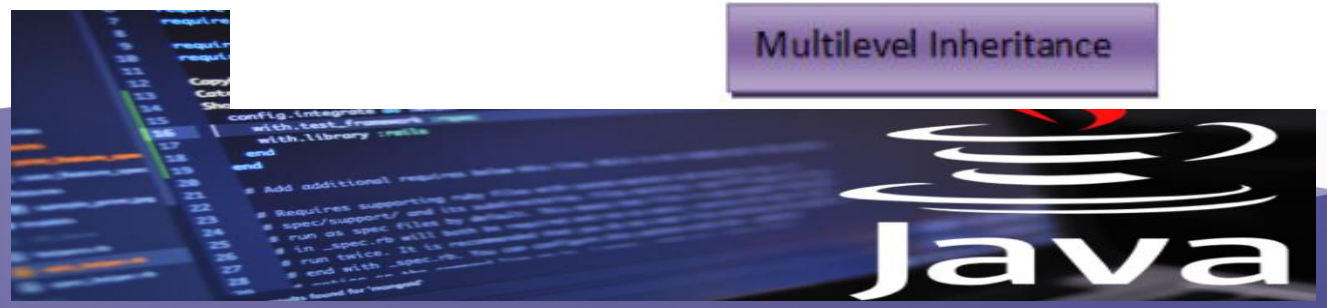
- **Class:** Class is a set of objects which shares common characteristics/ behavior and common properties/ attributes. Class is not a real-world entity. It is just a template or blueprint or prototype from which objects are created.
- **Super Class/Parent Class:** The class whose features are inherited is known as a superclass(or a base class or a parent class).
- **Sub Class/Child Class:** The class that inherits the other class is known as a subclass(or a derived class, extended class, or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.
- **Reusability:** Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.



Inheritance in Java

Terms that we need to get familiar with to really understand inheritance

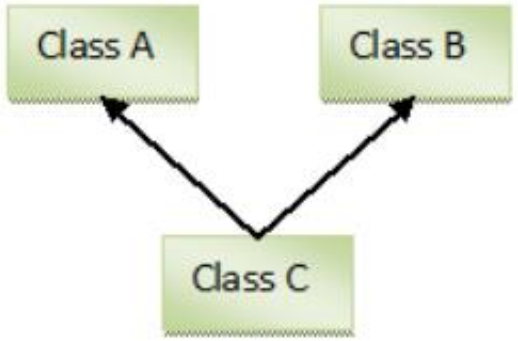




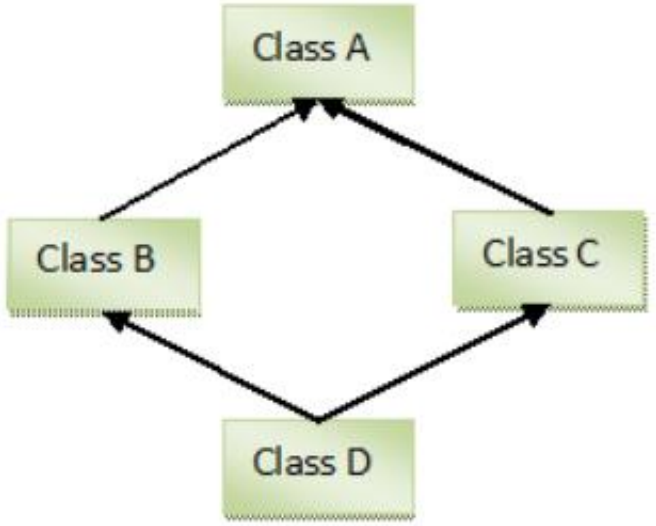
Inheritance in Java

Terms that we need to get familiar with to really understand inheritance

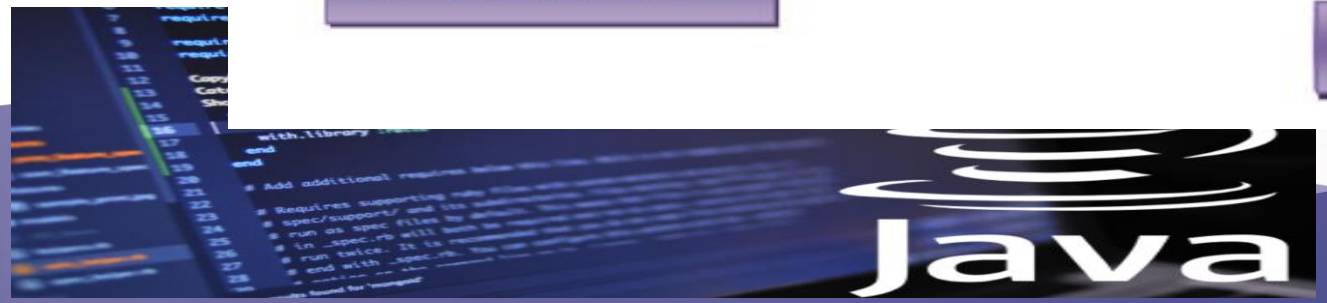
Multiple inheritance is not supported in java.



Multiple Inheritance

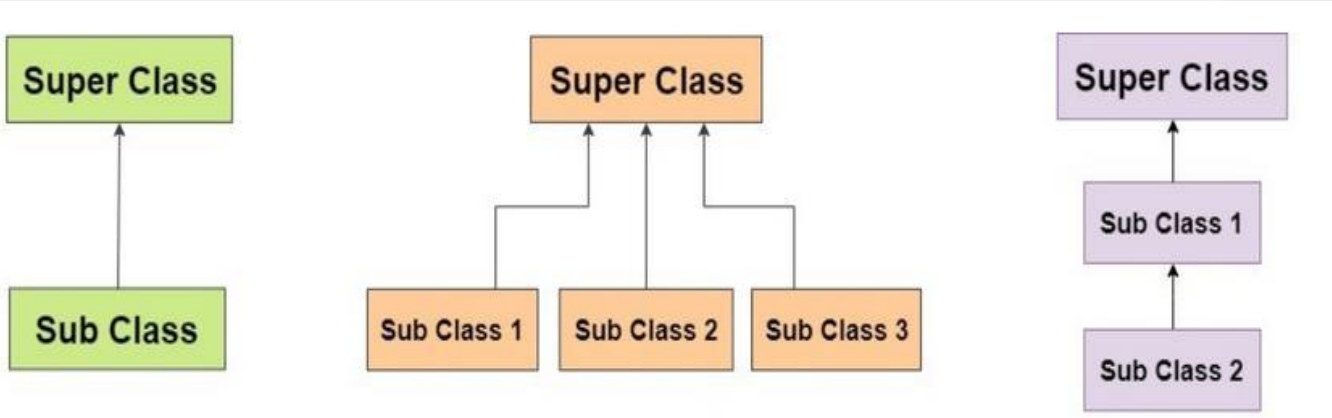


Hybrid Inheritance



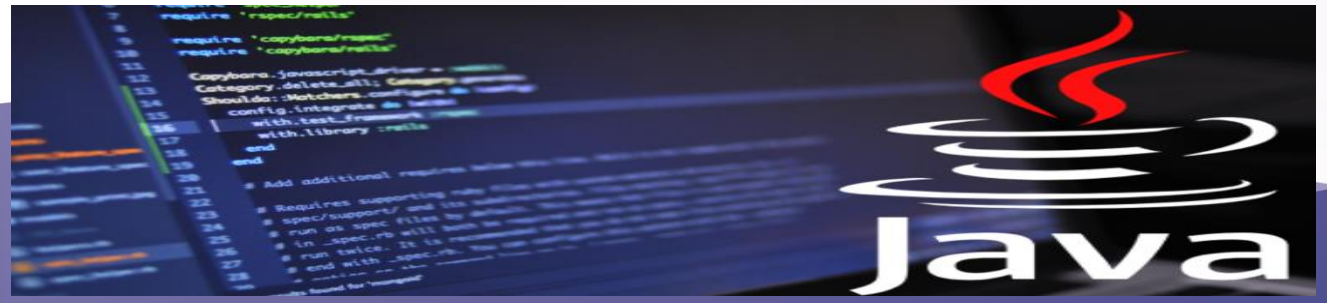
Inheritance in Java

Terms that we need to get familiar with to really understand inheritance



Super Class also called Parent class

Sub Class also called Child class



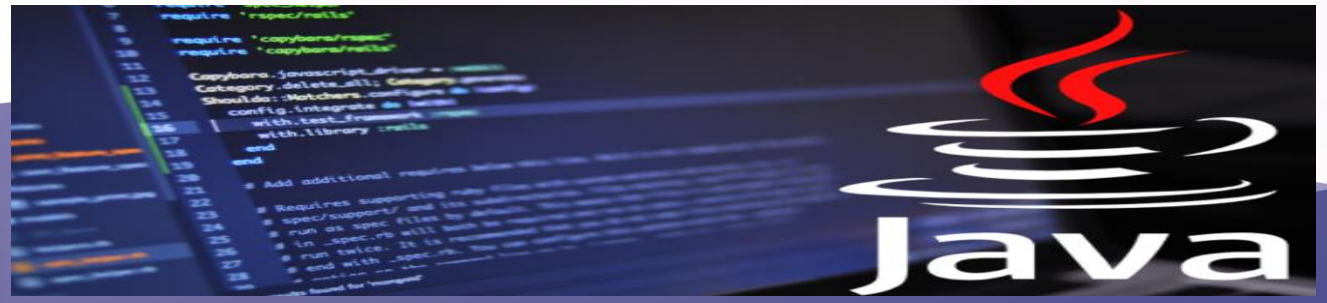
Inheritance in Java

This table shows the connection between Access Qualifiers and Containers in java (that is, Class, Package, Subclass, World)

Access Levels

Modifier	Class	Package	Subclass	World
`public`	Y	Y	Y	Y
`protected`	Y	Y	Y	N
no modifier	Y	Y	N	N
`private`	Y	N	N	N

NOTE : the restriction increases from Class to World …

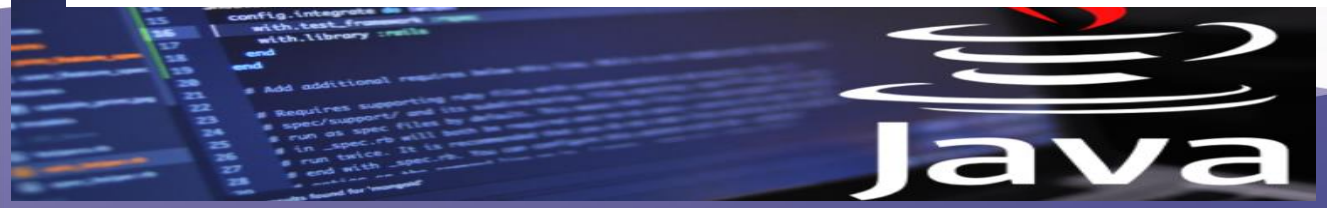


Inheritance in Java

The syntax of Java Inheritance

```
class Subclass-name extends Superclass-name
{
   //methods and fields
}
```

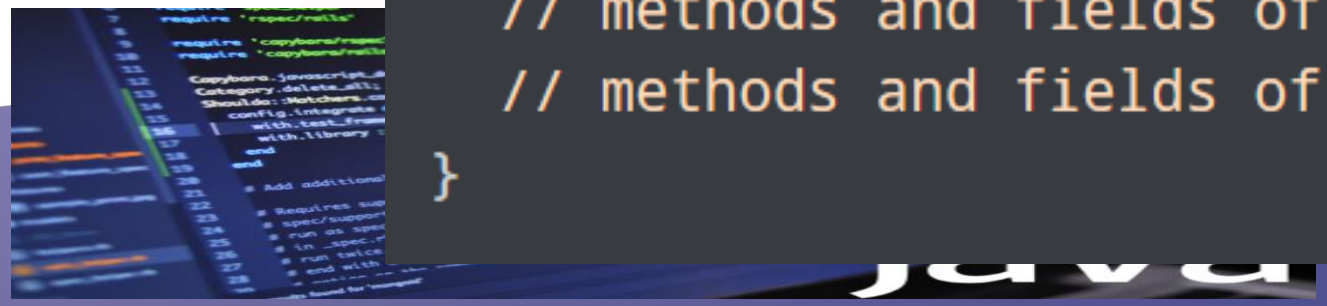
The **extends keyword** indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.



Inheritance in Java

Example

```
class Animal {
  // methods and fields
}

// use of extends keyword
// to perform inheritance
class Dog extends Animal {

  // methods and fields of Animal
  // methods and fields of Dog
}
```

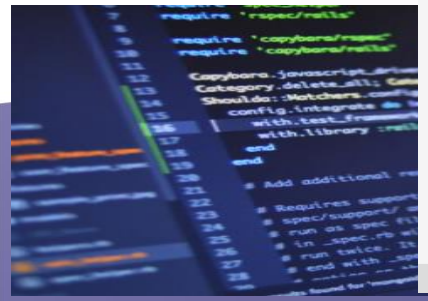


Inheritance in Java

Example

```java
class Animal {

  // field and method of the parent class
  String name;
  public void eat() {
    System.out.println("I can eat");
  }
}

// inherit from Animal
class Dog extends Animal {

  // new method in subclass
  public void display() {
    System.out.println("My name is " + name);
  }
}

class Main {
  public static void main(String[] args) {

    // create an object of the subclass
    Dog labrador = new Dog();

    // access field of superclass
    labrador.name = "Rohu";
    labrador.display();

    // call method of superclass
    // using object of subclass
    labrador.eat();

  }
}
```

```
My name is Rohu
I can eat
```



Inheritance in Java

To summarize

1) What is Inheritance ?
2) When do you use it ?
3) How do you implement Inheritance in Java ?