INHERITANCE:- Inheritance in JAVA is a mechanism in which one object acquires all the properties and behaviour of a parent object.

IDEA:-

The idea behind inheritance in JAVA is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class.

*. Inheritance represents the IS-A relationship which is also known as a parent-child relationship.

WHY USE:-

- For Method Overiding
- For Code Reusability

*. Inheritance is one of the cornerstones of OOP because it allows the creation of hierarchical classification.

→ In the terminology OF JAVA, a class that is inherited is called a superclass. The class that does the inheriting is called a subclass. Therefore, a subclass is a specialized version OF a superclass. It inherits all OF the members defined by the superclass and add its own, unique elements.

To inherit a class, you simply incorporate the definition OF one class into another by using the extends keyword.

TERMS USED IN INHERITANCE:

- CLASS
- SUPER CLASS/PARENT CLASS:- Superclass is the class From where a subclass inherits the Feature. It is also called a base class or a parent class.

- **SUB CLASS/CHILD CLASS:-** Subclass is a class which inherits the base class. It is also called a derived class, extend class, or child class.

- **REUSABILITY**

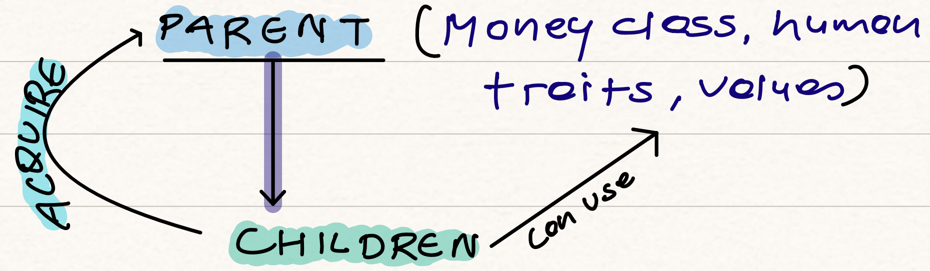
SYNTAX:-

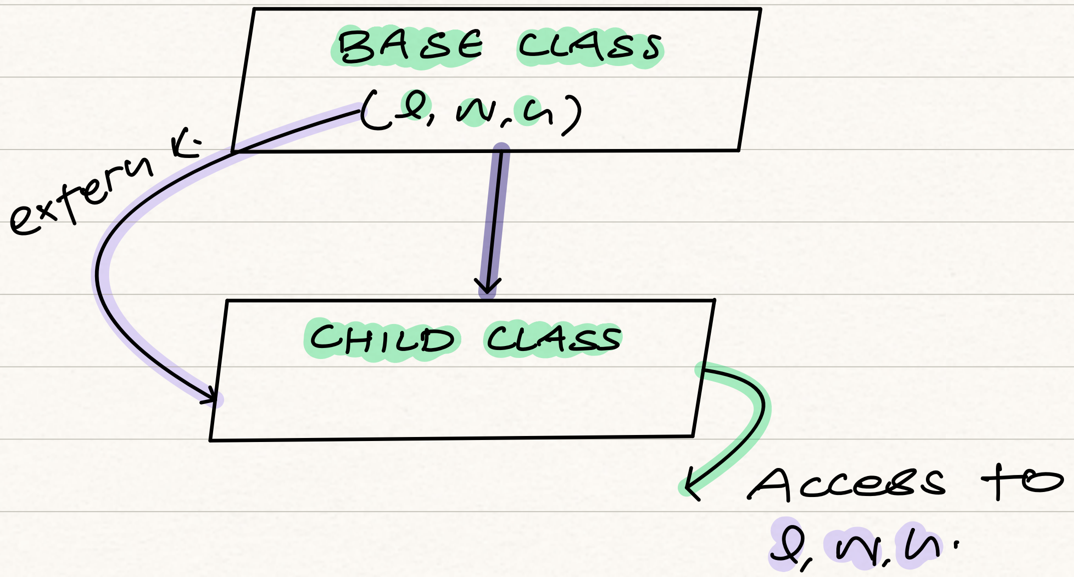
```
class Subclass-name extends Superclass-name
{
    // method & Fields
}
```

WHERE:-

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

EXAMPLE:-


PARENT
(Money class, human traits, values)
ACQUIRE
CHILDREN
can use


BASE CLASS
(l, w, h)
extend:
CHILD CLASS
Access to
l, w, h.

PROGRAM:-

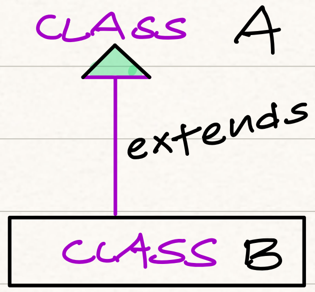
```
        class Employee {
            int salary = 65000;
          }

    class Accountant extends Employee
                  {
            int bonus = 25000 :
public static void main (String[] args)
                  {

  Accountant devraj = new Accountant();
   System.out.println(" Accountant
                  Salary is: "+devraj.
                                salary);
   System.out.println(" Bonus = "+
                       devraj.bonus);
          }
```

TYPES OF INHERITANCE:-

1. Single Inheritance
2. Multiple Inheritance
3. Multi-level Inheritance
4. Hierarchical Inheritance
5. Hybrid Inheritance

1 SINGLE INHERITANCE:- A single subclass extends from a single superclass.

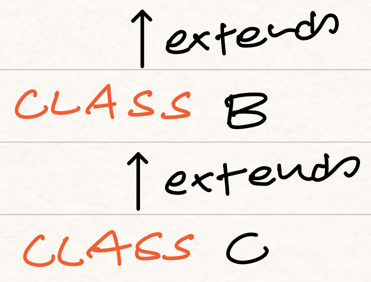
FOR EXAMPLE:-



2 MULTILEVEL INHERITANCE:- A subclass extend from a superclass and then the same subclass acts as a superclass for another class.

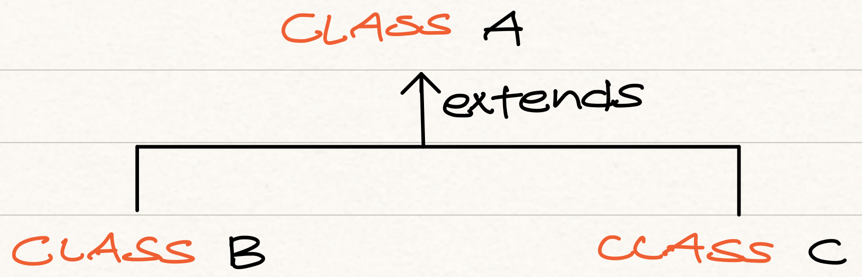
FOR EXAMPLE:-

CLASS A



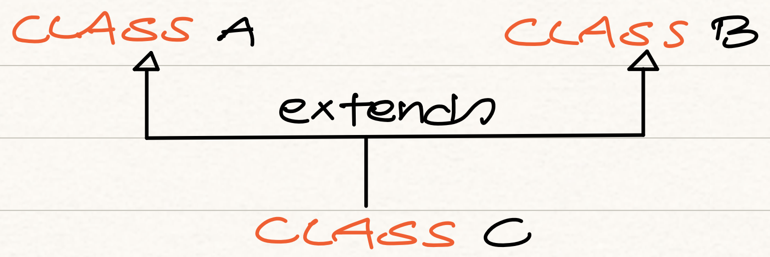
③ **HIERARCHICAL INHERITANCE** :- Multiple subclasses extend From a single superclass.

FOR EXAMPLE:-



④ **MULTIPLE INHERITANCE** :- A single subclass extends From multiple superclasses.

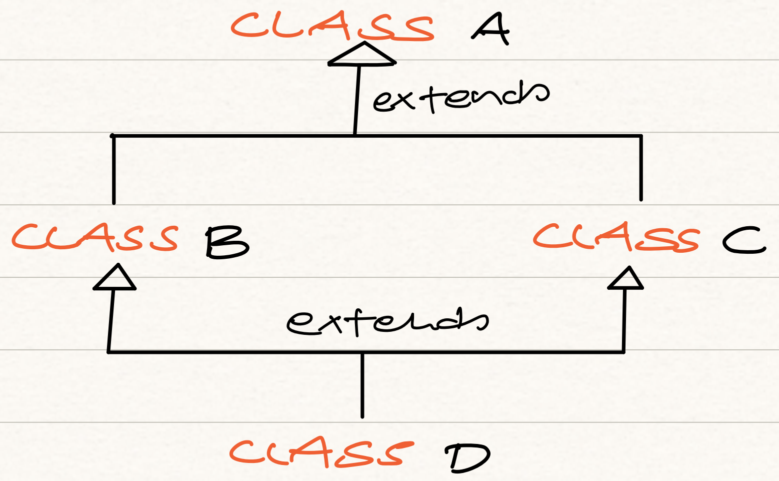
FOR EXAMPLE:-



NOTE:- Java doesn't support multiple inheritance. However, we can achieve multiple inheritance using interfaces.

5. HYBRID INHERITANCE:- Hybrid Inheritance is a combination of two or more types of inheritance.

FOR EXAMPLE:-



WHY MULTIPLE INHERITANCE IS NOT SUPPORTED IN JAVA ?

→ To reduce the complexity and simplify the language, multiple inheritance is not supported in Java.

DIAMOND PROBLEM

SCENARIO: A, B and C are three classes. The C class inherits A and B classes. If A and B classes have the same method and you call it from child class object, there will be ambiguity to call the method of A or B class.

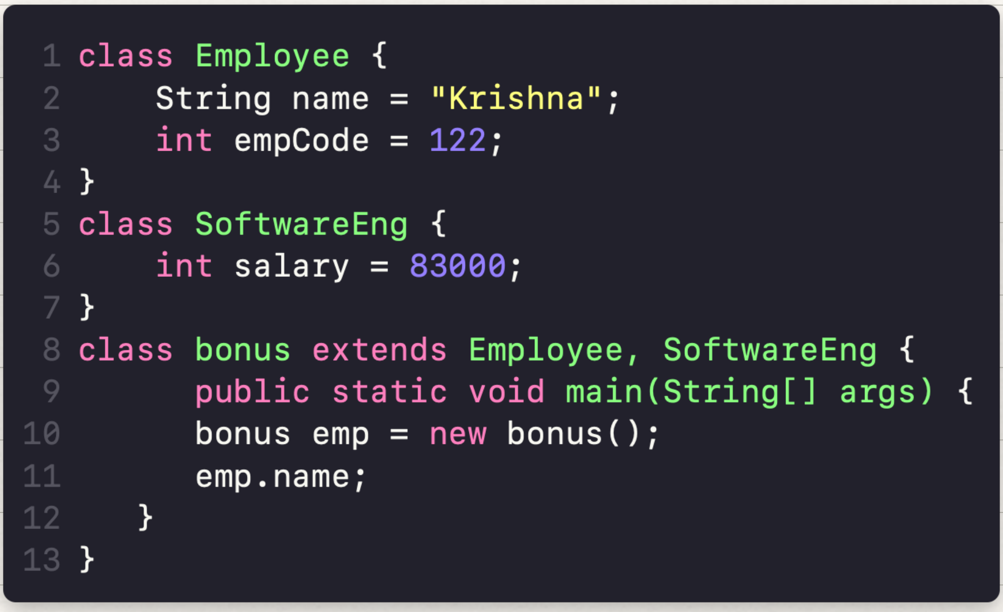
Since,

Compile-time errors are better than run-time errors. Java renders compile-time error if you inherit 2 classes.

So,

whether you have some method or different, there will be compile-time error.

PROGRAM :-



```java
class Employee {
    String name = "Krishna";
    int empCode = 122;
}
class SoftwareEng {
    int salary = 83000;
}
class bonus extends Employee, SoftwareEng {
      public static void main(String[] args) {
      bonus emp = new bonus();
      emp.name;
   }
}
```