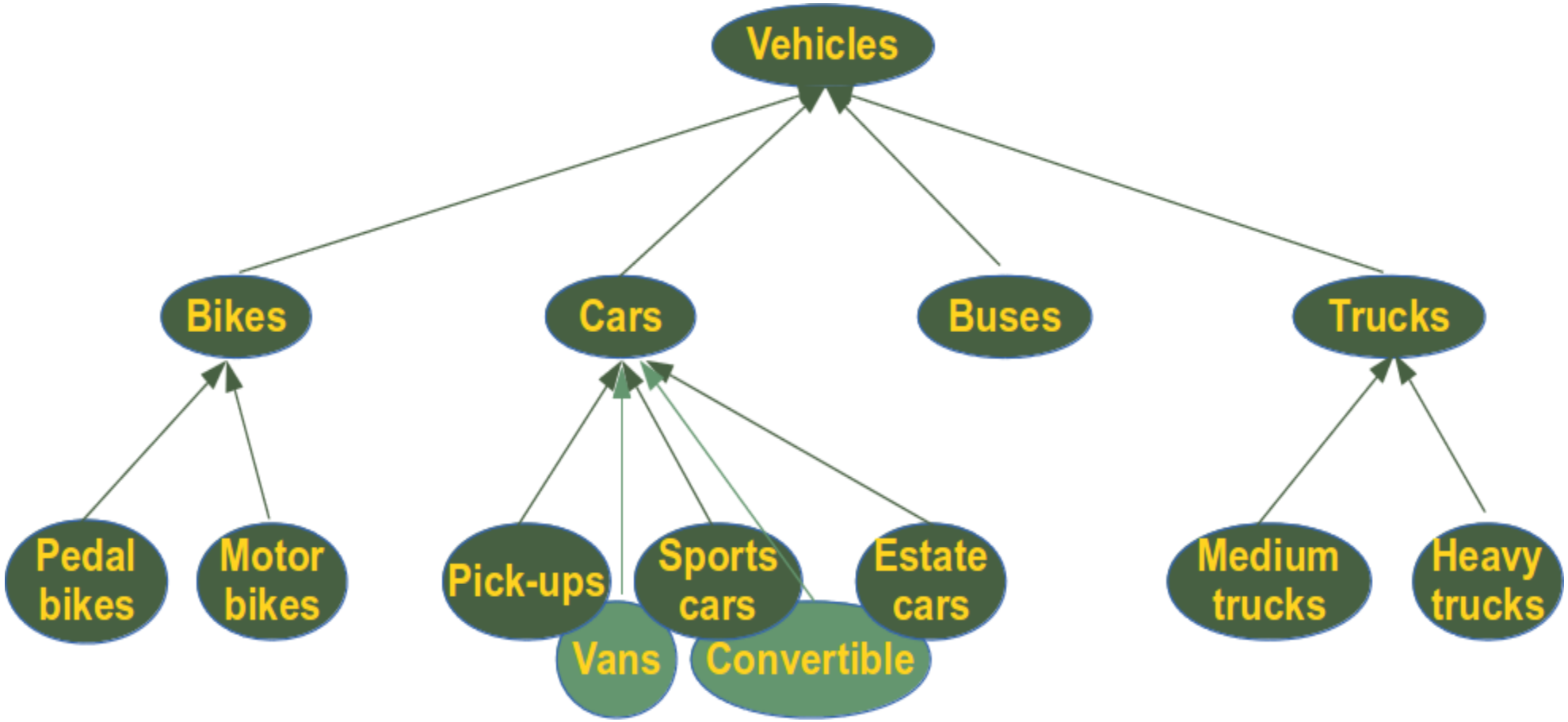
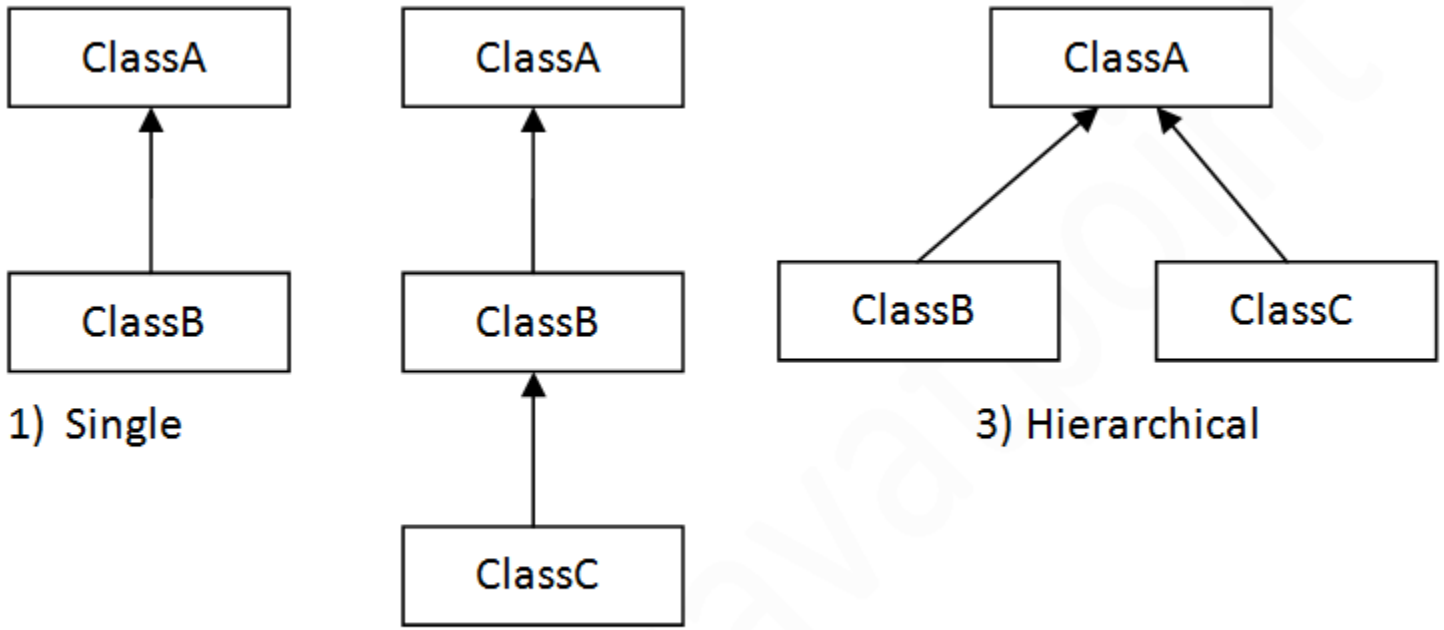
Inheritance

- Inheritance is one of the feature of Object-Oriented Programming. Inheritance allows a class to use the properties and methods of another class.
- The derived class is also called subclass and the base class is also known as super-class
- A super-class can have any number of subclasses. But a subclass can have only one superclass.
- Java does not support multiple inheritance.

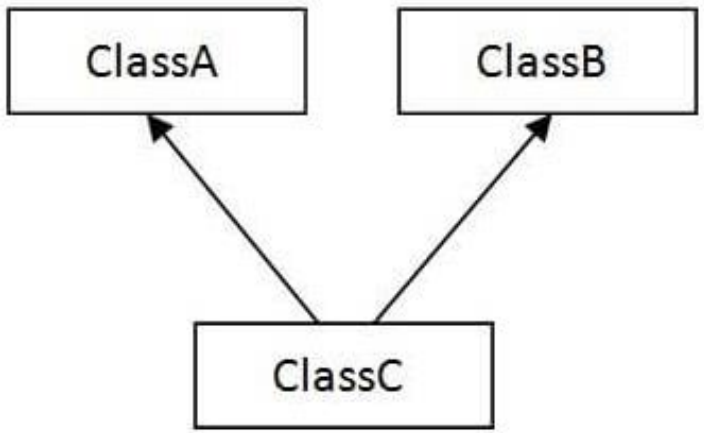
Inheritance



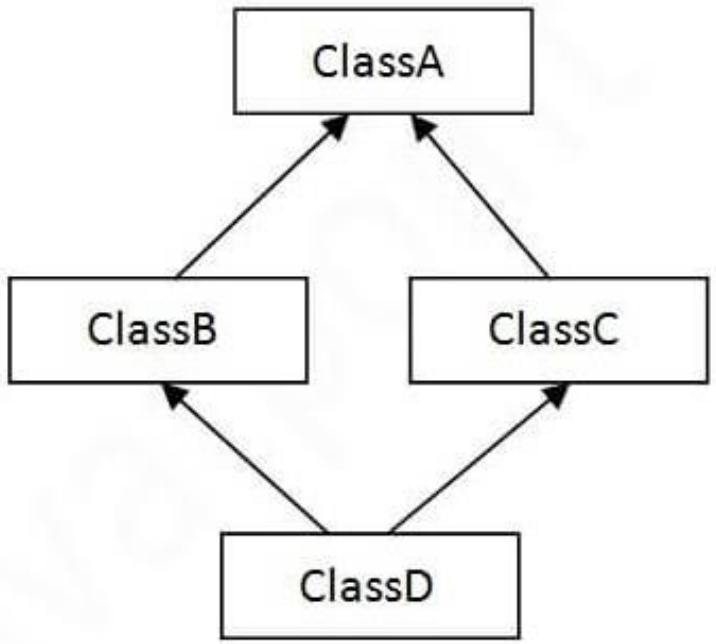
Types of inheritance in java



Continued…



4) Multiple



5) Hybrid

inheritance

- The keyword used for inheritance is extends.

```
public class ChildClass extends BaseClass  {

   // derived class methods extend and possibly override

}
```

Java Inheritance Example

```
class Employee{
 float salary=40000;
}
class Programmer extends Employee{
 int bonus=10000;
}
public static void main(String args[]){
   Programmer p=new Programmer();
   System.out.println("Programmer salary is:"+p.salary);
   System.out.println("Bonus of Programmer is:"+p.bonus);
}
```

Single Inheritance Example

```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class TestInheritance{
public static void main(String args[]){
Dog d=new Dog();
d.bark();
d.eat();
}}
```

Output:

```
barking...
eating...
```

Multilevel Inheritance Example

```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class BabyDog extends Dog{
void weep(){System.out.println("weeping...");}
}
class TestInheritance2{
public static void main(String args[]){
BabyDog d=new BabyDog();
d.weep();
d.bark();
d.eat();
}}
```

Output:

```
weeping...
barking...
eating...
```

Hierarchical Inheritance Example

```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class Cat extends Animal{
void meow(){System.out.println("meowing...");}
}
class TestInheritance3{
public static void main(String args[]){
Cat c=new Cat();
c.meow();
c.eat();
//c.bark();//C.T.Error
}}
```

Output:

```
meowing...
eating...
```

Why multiple inheritance is not supported in java?

- To reduce the complexity and simplify the language, multiple inheritance is not supported in java.
- Consider a scenario where 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 runtime errors, Java renders compile-time error if you inherit 2 classes. So whether you have same method or different, there will be compile time error.

Continued…

- **class** A{
- **void** msg(){System.out.println("Hello");}
- }
- **class** B{
- **void** msg(){System.out.println("Welcome");}
- }
- **class** C **extends** A,B{//suppose if it were
-
- **public static void** main(String args[]){
- C obj=**new** C();
- obj.msg();//Now which msg() method would be invoked?
- }
- }