

CONTENTS

CHAPTER	TOPICS	Page No
1.	INTRODUCTION TO JAVA	1
2.	BASIC SYNTAX & OOPS CONCEPT	7
3.	DATA TYPE, VARIABLE & CONSTANT	12
4.	RUN TIME INPUT	17
5.	LOOP STATEMENT	23
6.	DECISION CONTROL STATEMENT	26
7.	FUNCTIONS	32
8.	CLASS & OBJECTS	37
9.	CONSTRUCTOR & DESTRUCTOR	43
10.	INHERITANCE	47
11.	STATIC	51
12.	FINAL	56
13.	ABSTRACTION	60
14.	POLYMORPHISM	65
15.	INTERFACE	71
16.	PACKAGES	79

INHERITANCE

Inheritance

One of the most important concepts in object-oriented programming is that of inheritance. Inheritance is a concept of linking two or more classes with each other in a hierarchical manner so that their properties and functions can be shared. (One class will extend to another class.)

When you want to create a new class and there is already a class that includes some of the code that you want, you can derive your new class from the existing class. In doing this, you can reuse the fields and methods of the existing class without having to write them again.

Advantages of Inheritance are:

- Code reusability
- Saves time
- Functionality of the program can be enhanced quickly
- Increases the programs reliability

Base class and Derived class:

A class that is derived from another class is called a **subclass** (also a **derived class**, extended class, or child class).

The class from which the subclass is derived is called a **superclass** (also a **base class** or a parent class).

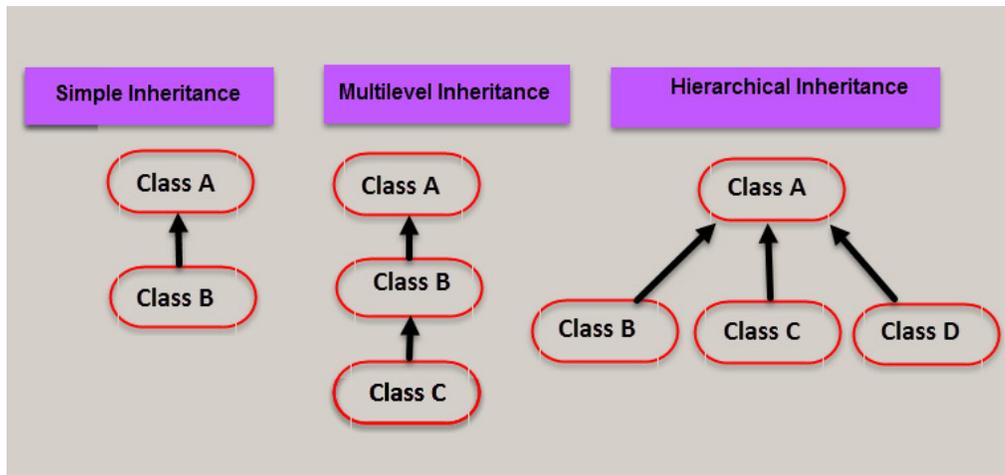
The inheritance relationship enables a derived class to inherit all the features from its base class and in addition to it new features can also be added. Any number of classes can be inherited from a base class. When a class is derived from another class, the derived class can access the data members and the member functions of the base class.

Types of Inheritance:

Java supports three different types of inheritance.

- Simple/Single
- Multilevel
- Hierarchical

Note:- To reduce the complexity and simplify the language, multiple inheritance is not supported in java.



If you want to derive a subclass from a superclass, you can do it as follows:

```
class <subclass name> extends <superclass name> { ... }
```

Some characteristics to be noted while using inheritance are:-

- Private members of the superclass are not inherited by the subclass.
- Constructors and initializer blocks are not inherited by a subclass.
- A subclass can extend only one superclass

Let us see an example of simple inheritance. Consider the super class vehicles and the sub class car, the sub class car can inherit the properties the properties of vehicle. With use of the extends keyword the subclasses will be able to inherit all the properties of the superclass except for the private properties of the superclass.

File 1: vehicle.java

```
import java.util.Scanner;  
  
public class vehicle {  
    String type;  
    int noOfTyres,noOfPassengers ;  
    Scanner sysin=new Scanner(System.in);  
    public void title(){
```

```
        System.out.println("*****VEHICLE*****");
    }
    private void vehicleDetails(){
        type=sysin.next();
        System.out.println("type "+type);
    }
}
```

File 2: cars.java

```
public class cars extends vehicle {
    public void subTitle()
    {
        System.out.println("*****CARS*****");
    }
    public void carDetails(){
        System.out.print("Enter No Of Tyres ");
        noOfTyres=sysin.nextInt();
        System.out.println("No Of Tyres "+noOfTyres);
        System.out.print("Enter No Of Passengers ");
        noOfPassengers=sysin.nextInt();
        System.out.println("No Of Passengers "+noOfPassengers);
    }
    public static void main(String[] args) {
        cars c1= new cars();
        c1.title();           //public method inherited from superclass
        c1.vehicleDetails();
        // vehicle details is a private method in super class cannot be inherited
        c1.subTitle();
        c1.carDetails();
    }
}

OUTPUT:-
*****VEHICLE*****
*****CARS*****
Enter No Of Tyres 4
No Of Tyres 4
Enter No Of Passengers 5
No Of Passengers 5
```

Multilevel Inheritance:

You can also extend this car class further, based on the type of car say suv, convertible, sports cars etc., or based on the fuel used. Here the class suv extends class cars to inherit the properties of cars class.

```
public class suv extends cars{
    public void subTitle()
    {
        super.subTitle();
        System.out.println("*****SUV*****");
    }
    public static void main(String[] args) {
        suv s1= new suv();
        s1.title();
        s1.subTitle();
        s1.carDetails();
    }
}
```

OUTPUT:

```
*****VEHICLE*****
*****CARS*****
*****SUV*****
Enter No Of Tyres 4
No Of Tyres 4
Enter No Of Passengers 7
No Of Passengers 7
```

The keyword **super** used in the subTitle() method is used to access the member function in the super class. You can also use the keyword super to access the data members of super class.