**Inheritance In C# Programming**

* The similarity in physical features of a child to that of its parents is due to the child having inherited these features from its parents.
* Similarly, in C#, inheritance allows you to create a class by deriving the common attributes and methods of an existing class.
* Inheritance provides reusability by allowing us to extend an existing class.
* The reason behind OOP programming is to promote the reusability of code and to reduce complexity in code and it is possible by using inheritance.

**EMPLOYEE**

**Int EmpId;**

**String EmpName;**

**Int EmpAge;**

**Int EmpContactNo;**

**Parent Class**

**Child Class**

**Child Class**

**PERMANENT EMPLOYEES**

**Int EmpId;**

**String EmpName;**

**Int EmpAge;**

**Int EmpContactNo;**

**Int permanentSalary;**

**Int permanentHours;**

**VISITING EMPLOYEES**

**Int EmpId;**

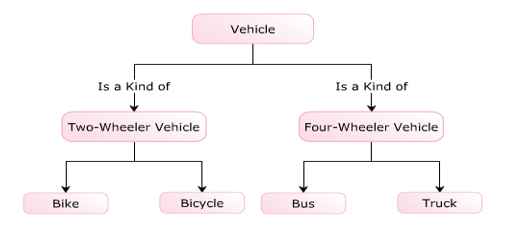
**String EmpName;**

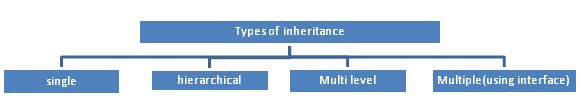
**Int EmpAge;**

**Int EmpContactNo;**

**Int visitingSalary;**

**Int visitingHours;**



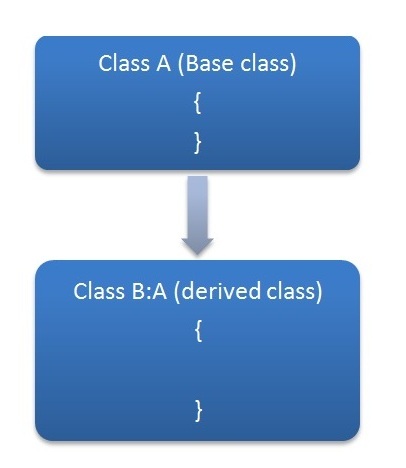
* The following are the types of inheritance in C#.

The inheritance concept is based on a base class and derived class. Let us see the definition of a base and derived class.

* **BASE CLASS** - is the class from which features are to be inherited into another class.
* **DERIVED CLASS** - it is the class in which the base class features are inherited.

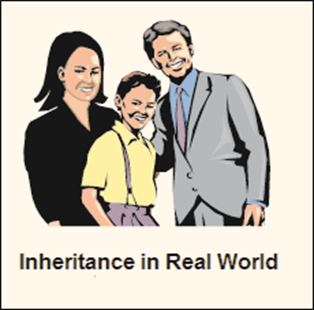
**Single Inheritance**

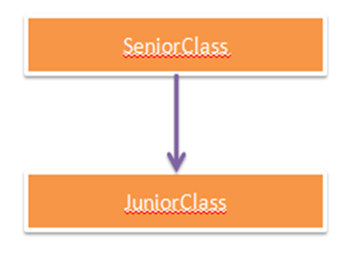
It is the type of inheritance in which there is one base class and one derived class.



**Definition of Inheritance**

* The similarity in physical features of a child to that of its parents is due to the child having inherited these features from its parents.
* Similarly, in C#, inheritance allows you to create a class by deriving the common attributes and methods of an existing class.
* The class from which the new class is created is known as the base class and the created class is known as the   
  derived class.
* The process of creating a new class by extending some features of an existing class is known as inheritance.

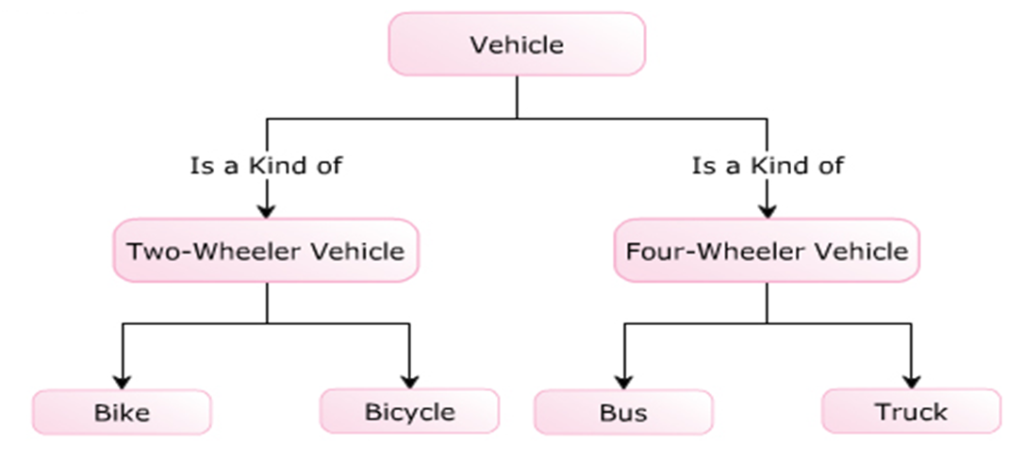




**Example**

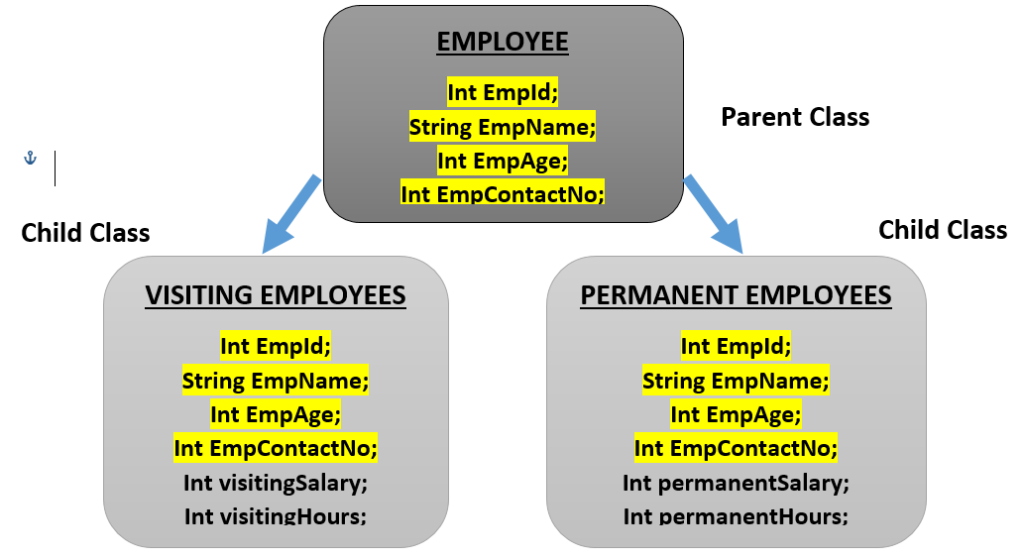
* Consider a class called Vehicle that consists of a variable called color and a method called Speed().
* These data members of the Vehicle class can be inherited by the TwoWheelerVehicle and FourWheelerVehicle classes.

**The following figure illustrates an example of inheritance:**



**Inheritance In C#**

* The similarity in physical features of a child to that of its parents is due to the child having inherited these features from its parents.
* Similarly, in C#, inheritance allows you to create a class by deriving the common attributes and methods of an existing class.
* Inheritance provides reusability by allowing us to extend an existing class.
* The reason behind OOP programming is to promote the reusability of code and to reduce complexity in code and it is possible by using inheritance.



**Purpose Of Inheritance**

* The purpose of inheritance is to reuse common methods and attributes among classes without recreating them.
* Reusability of a code enables you to use the same code in different applications with little or no changes.

**Example**

* Consider a class named Animal which defines attributes and behavior for animals.
* If a new class named Cat has to be created, it can be done based on Animal because a cat is also an animal.
* Thus, you can reuse the code from the   
  previously-defined class.



**Apart from reusability, inheritance is widely used for:**

1. Generalization
2. Specialization
3. Extension

**Generalization**

* Inheritance allows you to implement generalization by creating base classes. For example, consider the class Vehicle, which is the base class for its derived classes Truck and Bike.
* The class Vehicle consists of general attributes and methods that are implemented more specifically in the respective derived classes.

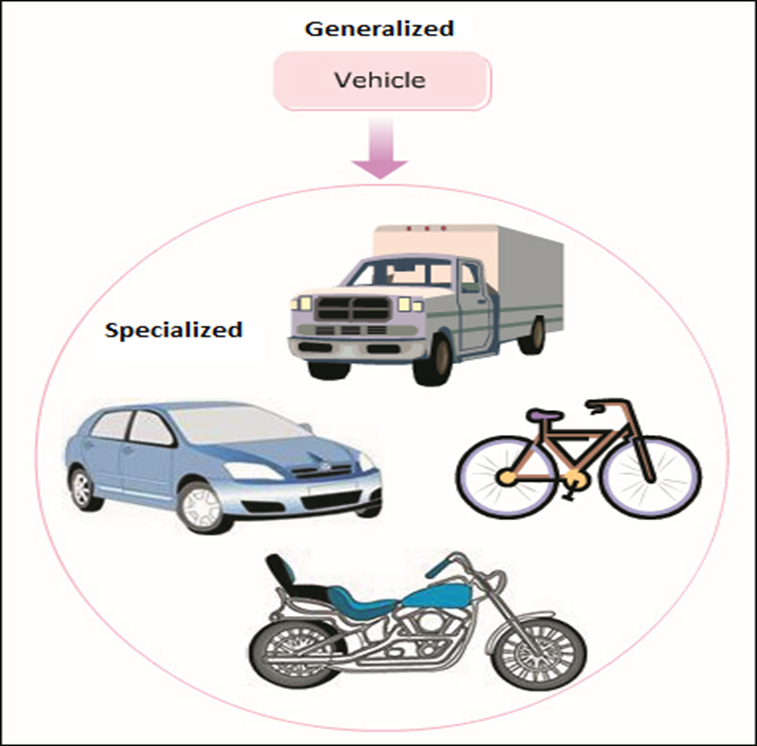
**Specialization**

* Inheritance allows you to implement specialization by creating derived classes.
* For example, the derived classes such as Bike, Bicycle, Bus, and Truck are specialized by implementing only specific methods from its generalized base class Vehicle.

**Extension**

* Inheritance allows you to extend the functionalities of a derived class by creating more methods and attributes that are not present in the base class. It allows you to provide additional features to the existing derived class without modifying the existing code.

**The following figure displays a real-world example demonstrating the purpose of inheritance:**



The inheritance concept is based on a base class and derived class. Let us see the definition of a base and derived class.

* **BASE CLASS -** is the class from which features are to be inherited into another class.
* **DERIVED CLASS -** it is the class in which the base class features are inherited.