Dart Inheritance:

Dart inheritance is defined as the process of deriving the properties and characteristics of another class. It provides the ability to create a new class from an existing class. It is the most essential concept of the oops(Object-Oriented programming approach). We can reuse the all the behavior and characteristics of the previous class in the new class.

- **Parent Class -** A class which is inherited by the other class is called **superclass** or **parent class**. It is also known as a **base class**.
- **Child Class -** A class which inherits properties from other class is called the child class. It is also known as the **derived class** or **subclass**.

Syntax -

```
class child_class extends parent_class {
   //body of child class
}
```

The child class inherits functions and variables, or properties of parent class using the extends keyword. It cannot inherit the parent class constructor; we will discuss this concept later.

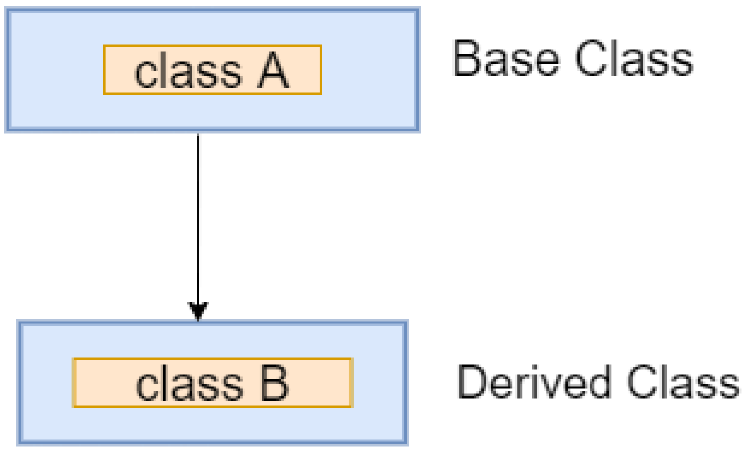
Types of Inheritance

The inheritance can be mainly four types. These are given below.

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance

Single Level Inheritance

In the single inheritance, a class is inherited by a single class or subclass is inherited by one parent class. In the following example, we create Person which inherits Human class.



Single Inhetitance

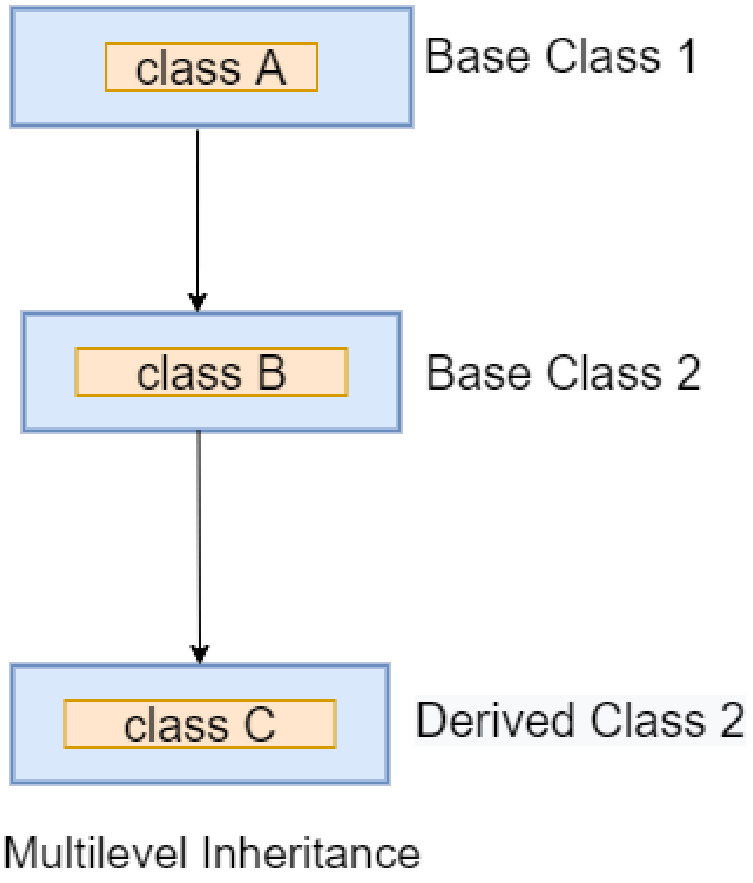
Let's understand the following example.

Example -

```
class Bird{
    void fly()
      {
        print("The bird can fly");
       }
  }
    // Inherits the super class
class Parrot extends Bird{
      //child class function
      void speak(){
         print("The parrot can speak");
            }
}
void main() {
    // Creating object of the child class
    Parrot p=new Parrot();
    p.speak();
    p.fly();
}
```

Multilevel Inheritance

In the multiple inheritance, a subclass is inherited by another subclass or creates the chaining of inheritance. Let's understand the following example.



Example -

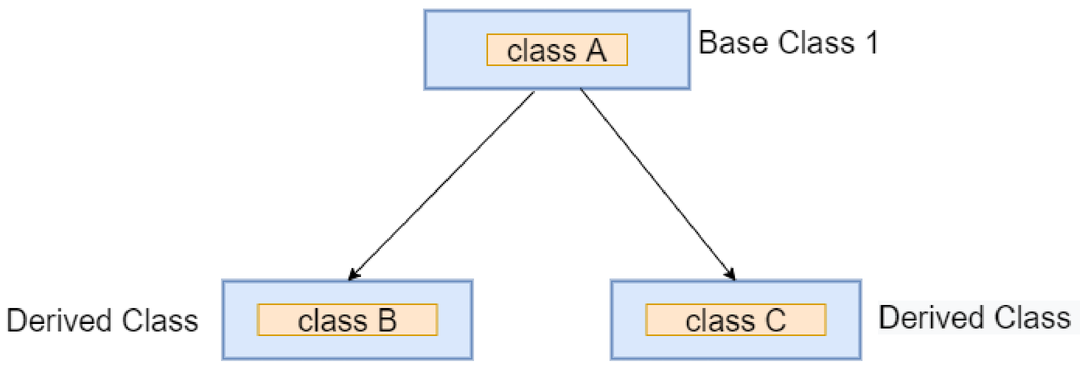
```
class Bird{
    void fly()
      {
        print("The bird can fly");
       }
  }
    // Inherits the super class
class Parrot extends Bird{
      void speak(){
         print("The parrot can speak");
            }

}

// Inherits the Parror base class
class Eagle extends Parrot {
        void vision(){
          print("The eagle has a sharp vision");
             }
}
void main() {
    // Creating object of the child class
    Eagle e=new Eagle();
    e.speak();
    e.fly();
    e.vision();
}
```

Hierarchical Inheritance

In the hierarchical inherence, two or more classes inherit a single class. In the following example, the two-child classes Peter and James inherit the Person class.



Hierarichal Inheritance

Example -

```
// Parent Class
class Person {
  void dispName(String name) {
    print(name);
  }

  void dispAge(int age) {
    print(age);
  }
}

class Peter extends Person {

  void dispBranch(String nationality) {
    print(nationality);
  }
}
//Derived class created from another derived class.
class James extends Person {
        void result(String result){
            print(result);
}
}
void main() {
    // Creating Object of James class
    James j = new James();
    j.dispName("James");
    j.dispAge(24);
    j.result("Passed");

  // Creating Object of Peter class
    Peter p = new Peter();
    p.dispName("Peter");
    p.dispAge(21);
    p.dispBranch("Computer Science");

}
```

Dart Super Constructor

The child class can inherit all properties (methods, variables) and behavior of parent expect parent class constructor.& The superclass constructor can be invoke in sub class by using the **super()** constructor. We can access both non-parameterized and parameterized constructor of superclass. Accessing the constructor of superclass is slightly different in the Dart. The syntax is given below.

Syntax:

```
SubClassConstructor():super() {
}
```

Implicit super

As we know that the constructor is automatically called when we instantiate a class. When we create the object of sub class, it invokes the constructor of sub class which implicitly invokes the parent class's default(non-parameterized) constructor. We can use **super()** constructor in our subclass to invoke superclass constructor. Let's understand the following example.

Example -

```
// Parent class
class Superclass {
        Superclass(){
            print("This is a superclass constructor");

             }
}
class Subclass extends Superclass
{
        Subclass(){
             print("This is a subclass constructor");
         }
         display(){
            print("Welcome to javatpoint");
}
}
void main(){
        print("Dart Implicit Superclass constructor call");
        // We create a object of sub class which will invoke subclass constr
uctor.
        // as well as parent class constructor.
        Subclass s = new Subclass();
        // Calling sub class method
        s.display();
}
```

Explicit super

If the superclass constructor consists of parameters then we require to call super() constructor with argument in to invoke superclass constructor in subclass explicitly. Let's understand the following example.

Example -

```
// Parent class
class Superclass {
        Superclass(String msg){
            print("This is a superclass constructor");
             print(msg);

             }
}
class Subclass extends Superclass
{
        Subclass():super("We are calling superclass constructor explicitly ")
{
             print("This is a subclass constructor");

         }
         display(){
            print("Welcome to javatpoint");
}
}
void main(){
        print("Dart Implicit Superclass constructor example");
        // We create an object of sub class which will invoke subclass constructor.
        // as well as parent class constructor.
        Subclass s = new Subclass();
        // Calling sub class method
        s.display();
}
```