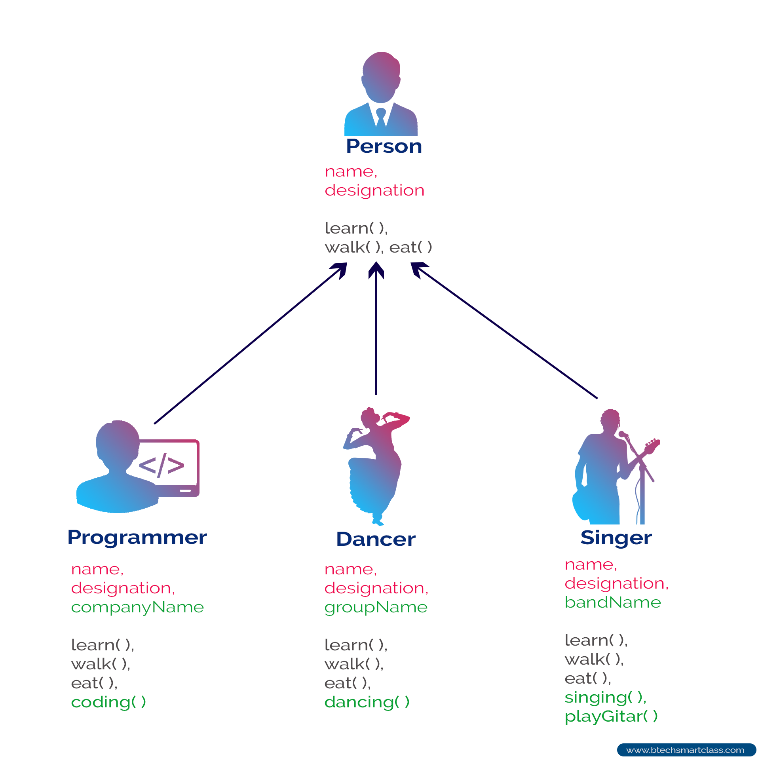
**Inheritance**

Inheritance is a mechanism   that allows one object to inherit all of the characteristics and actions of its parent object. Inheritance is one of the most significant characteristics of Object-Oriented Programming. The IS-A relationship, often known as a parent-child relationship, is represented by inheritance.

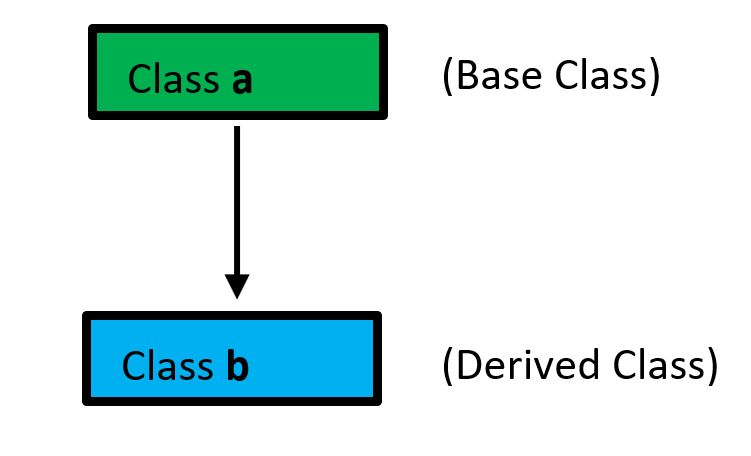


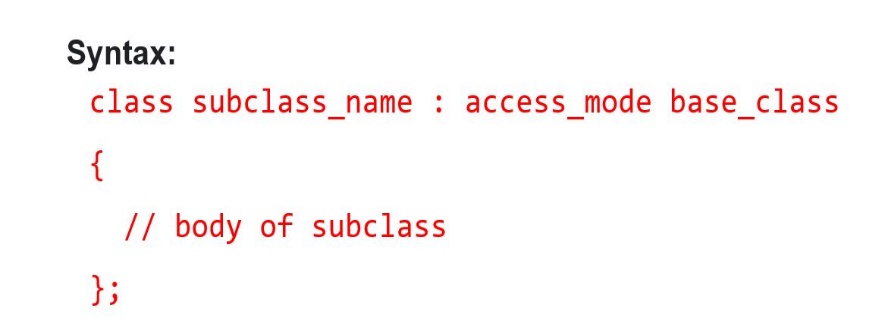
Inheritance has two types of Classes;

* **Subclass or Derived Class:** A Subclass or Derived Class is a class that inherits properties from another class.
* **Base Class or Superclass:** A Base Class or Superclass is a class whose properties are inherited by a subclass.

Inheritance can be divided in to five parts according to way that inherit the sub classes from parent class

1. Single inheritance : A class can only inherit from one other class in single inheritance. Only one base class inherits one subclass.

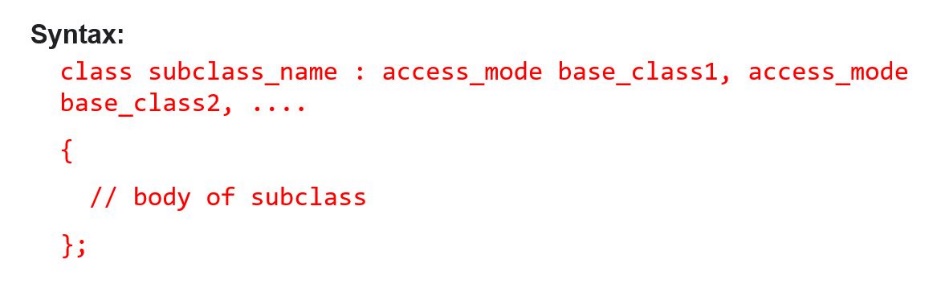


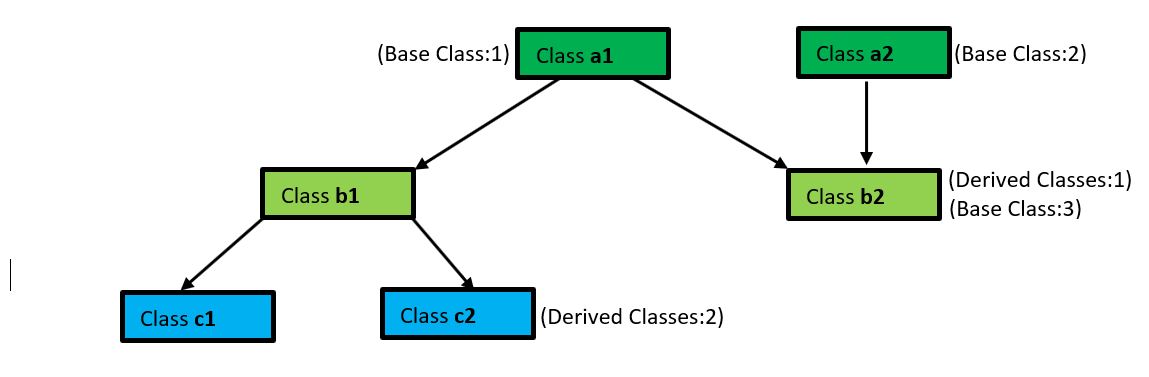


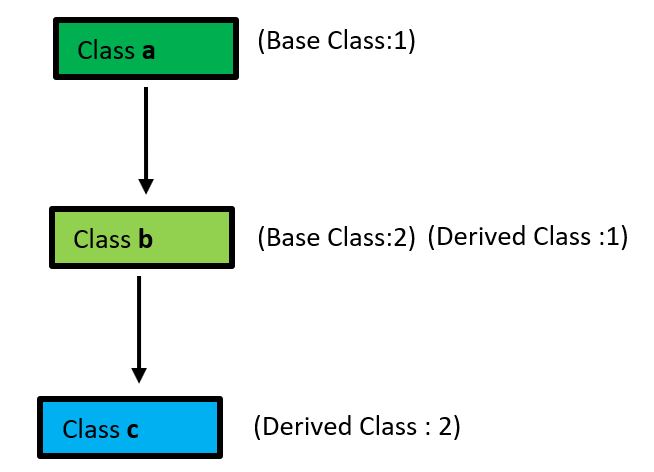
1. Multiple inheritance: Multiple Inheritance is a C++ feature that allows a class to inherit from multiple classes. More than one base class is inherited by a single subclass.

Diagram

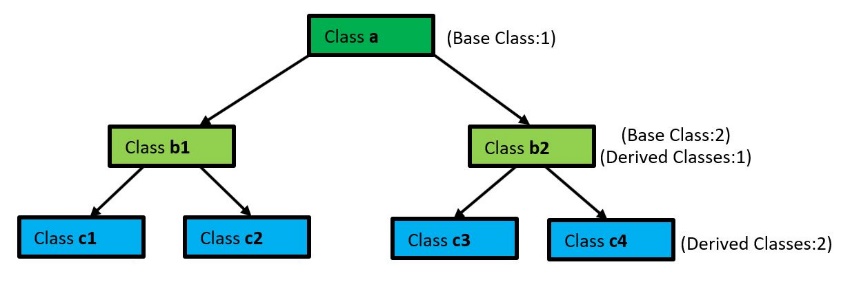
Description automatically generated

****

1. Multi-level inheritance: A derived class is produced from another derived class in this type of inheritance.



1. Hierarchical Inheritance: More than one subclass is inherited from a single base class in this style of inheritance. From a single base class, several derived classes are produced.



1. Hybrid Inheritance: Hybrid inheritance is created by merging multiple inheritance types. Combining Hierarchical Inheritance and Multiple Inheritance, for example.

**Access Modifiers which use in Inheritance**

Access modifiers specify which classes are allowed to use a particular attribute or method. There are 3 mode of access modifiers which using in Inheritance.

Diagram

Description automatically generated

**Public Method:**

If a subclass is derived from a public base class. The base class's public members will become public in the derived class, and the base class's protected members will become protected in the derived class.

**Protected Method:**

If a subclass is derived from a Protected base class. The base class's public and protected members will then become protected in the derived class.

**Private Method:**

If a subclass is derived from a Private base class. The base class's public and protected members will then become Private in the derived class.

**Default Method:**

Package-private is the default access modifier, which means that all members are viewable within the same package but not available from other packages.