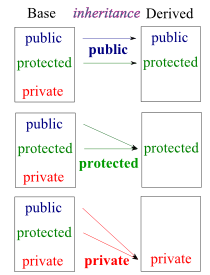
**Inheritance:**

**What ?**

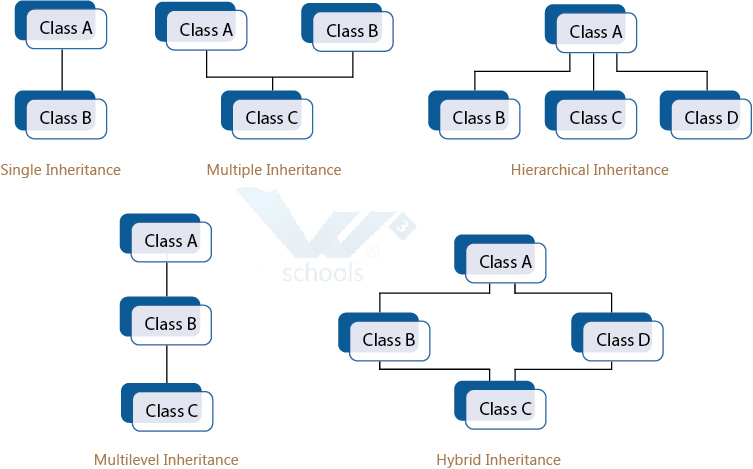
It is used to get the all the feature of the base class and have addition feature of its own in derived class.

A basic outline diagram about the inheritance on using with access specifier,



Through inheritance a user can create a new class from existing class.

Types of inheritance:



**Why ?**

1. Its used to combine the common functionality in a single class (base) and derived class are used to have a additional functionality which differentiates them from the base class.

2. Creating and maintaining the class is easier for inheritance.

**When ?**

When base class features are needed and additional functionality is needed to be added we go for inheritance.

**How ?**

*Program*

#include<iostream>

class Base

{

public:

Base()

{

std::cout << "Base Constructor\n";

}

~Base()

{

std::cout << "Base Destructor\n";

}

int fun()

{

std::cout << "Base::fun\n";

}

};

class Derived: public Base // Inheritance happens at this stage

{

public:

Derived()

{

std::cout << "Derived Constructor\n";

}

~Derived()

{

std::cout << "Derived Destructor\n";

}

int fun1()

{

std::cout << "Derived::fun1\n";

}

};

int main()

{

Derived d;

d.fun();

d.fun1();

return 0;

}

*Output:*

*Base Constructor*

*Derived Constructor*

*Base::fun*

*Derived::fun1*

*Derived Destructor*

*Base Destructor*