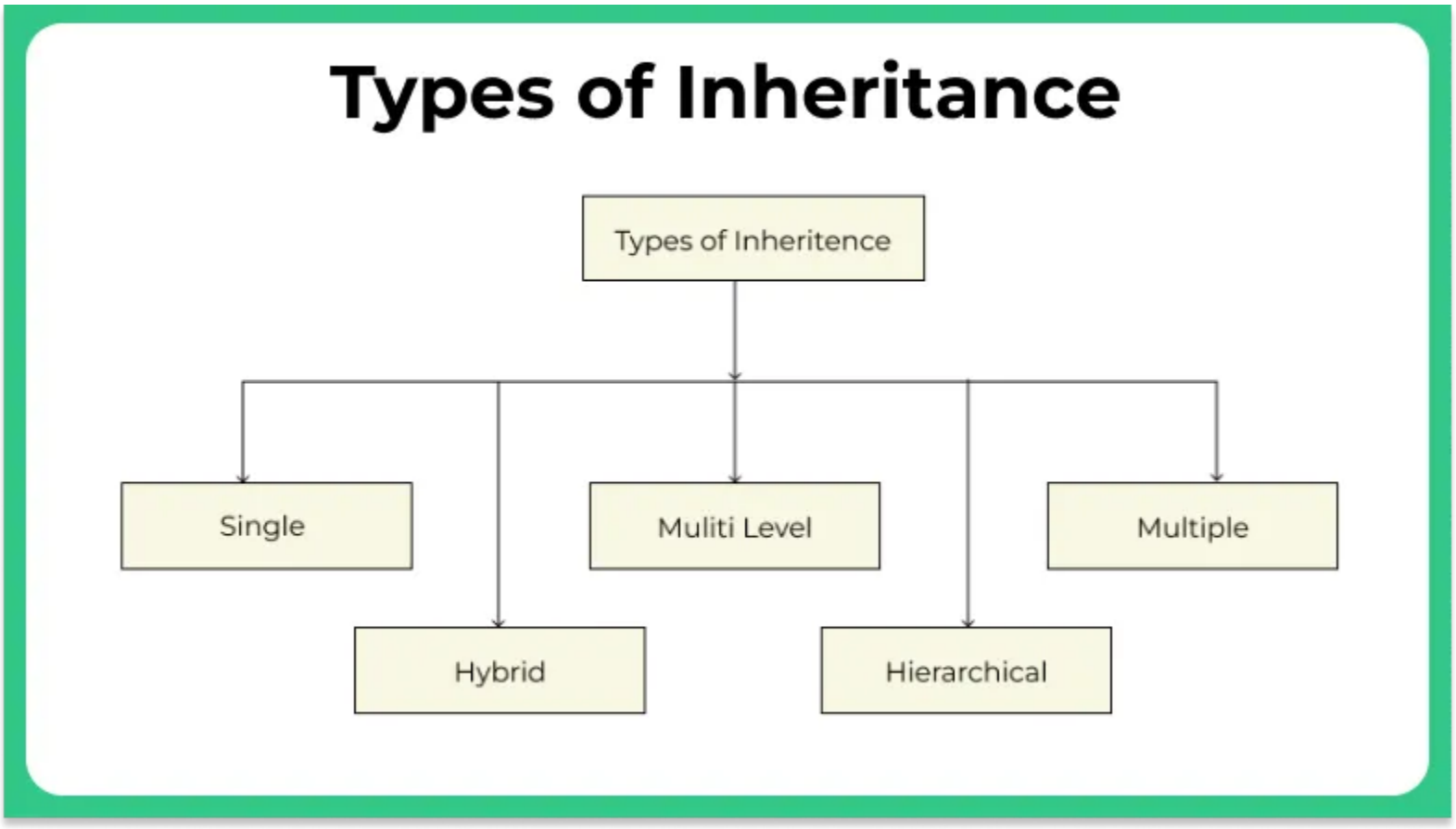
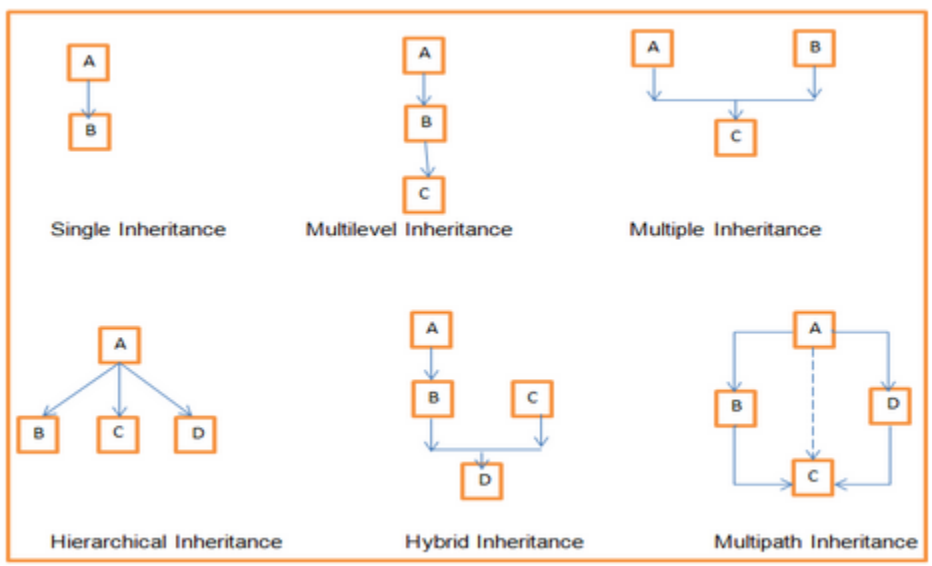
INHERITANCE

Types of inheritance:





Only public and protected variables and function inherited, private method or variable never inherited from base to derived class.

Single inheritance:

```
//Single class inheritance
//One base class (Parent class)  and one derived class (child class)

#include<iostream>
using namespace std;



//create base class
class A
{
	public:
		void print_A()
		{
			cout << endl << "Call method in class A";
		}
};

//Derived class from base class
class B : public A
{
	public:
		//Nothing inside class B (But all functionality of class A is available)

};

int main()
{
	//Create object
```

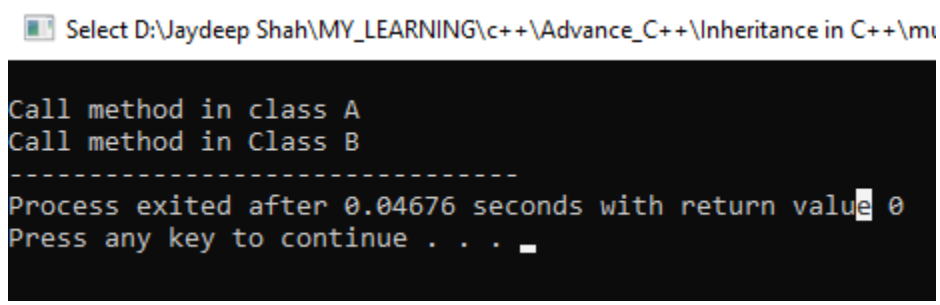
```
        B test;
        //call method of class A from object of class B.
        test.print_A();
        return 0;
}
```

Multiple Inheritance:

```
//Multiple class inheritance
//Multiple base class (Parent class)  and one derived class (child class)

#include<iostream>
using namespace std;

//create base class
class A
{
	public:
		void print_A()
		{
			cout << endl << "Call method in class A";
		}
};

//Other base class
class B
{
	public:
		void print_B()
		{
			cout << endl << "Call method in Class B";
		}

};
```

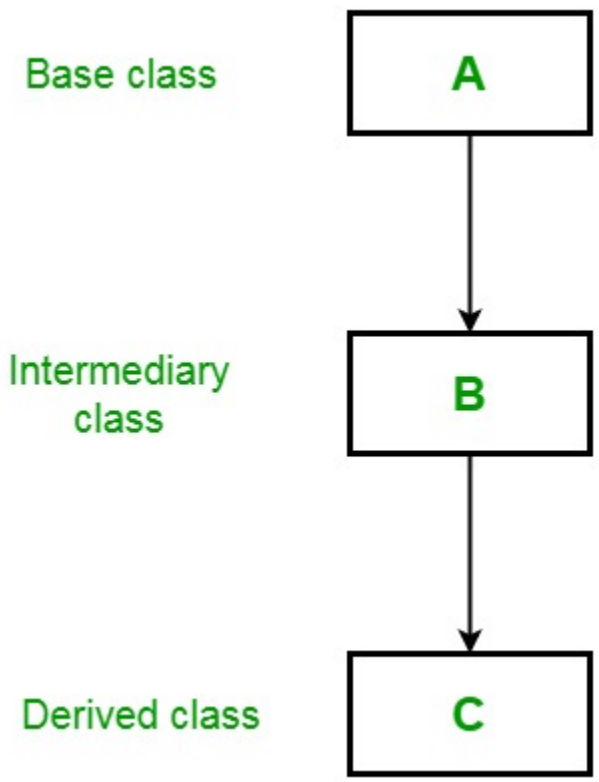
```
//Derived class(Multiple base class)
class C : public A, public B
{
	//Nothing inside C
};

int main()
{
	//Create object
	C test;
	//call method of class A from object of class C
	test.print_A();

	//call method of class B from object of class C
	test.print_B();

	return 0;
}
```

OUTPUT:



Multilevel inheritance:

Mullevel inheritance means one class derived from other derived class not from direct any based class.



Example code:

```
#include<iostream>
using namespace std;

//create base class
class A
{
	public:
		void print_A()
		{
			cout << endl << "Class A method call";
		}
};

//Derived class
class B : public A
```

```
{
	public:
		void print_B()
		{
			cout << endl << "Class B method call";
		}

};


//Multilevel inheritance
class C : public B
{
	public:
		//Nothing

};


int main()
{
	//Create object
	C test;
	//call method of class A from object of class C
	test.print_A();
	//call method of class B from object of class C
	test.print_B();

	return 0;
}
```

OUTPUT:

```
Class A method call
Class B method call
--------------------------------
Process exited after 0.04441 seconds with return value 0
Press any key to continue . . .
```