

What is Inheritance

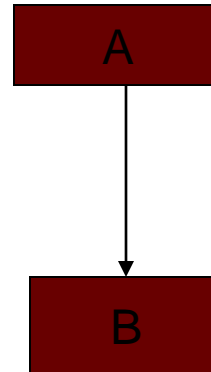
The mechanism of deriving a new class from an old one is called inheritance or derivation.

Definition of Inheritance:

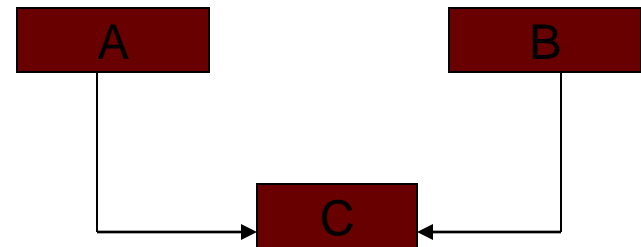
Inheritance is the mechanism which allows a class A to inherit properties of a class B.

Types of Inheritance:

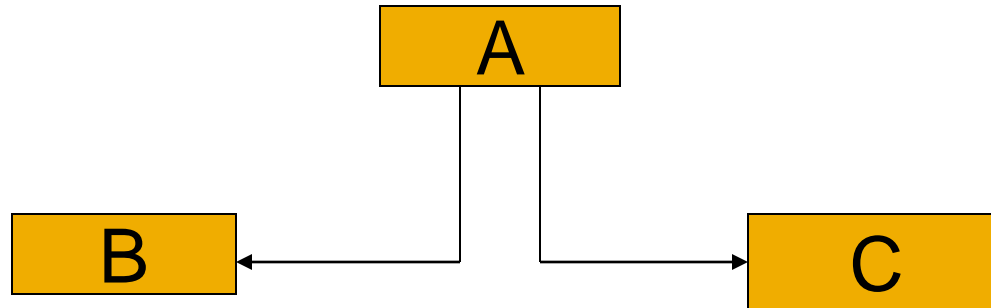
- **Single inheritance**



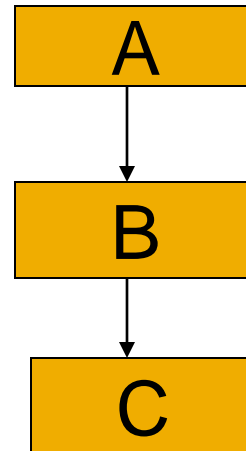
- **Multiple inheritance**



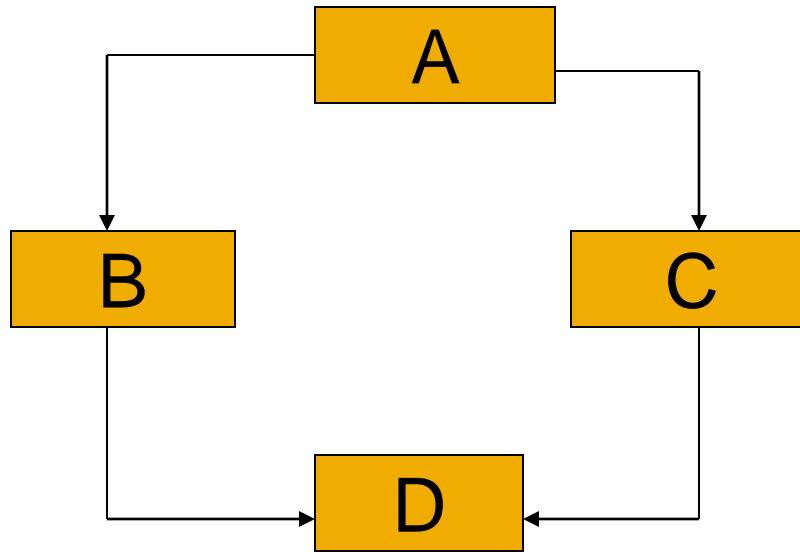
Hierachical Inheritance:



Multilevel Inheritance:



Hybrid Inheritance:



Advantages of Inheritance:

- The major advantage is the reusability. Once a base class is written and debugged, it need not be touched again, but can be used to work in different.
- It also reduces the frustration in complex programming.
- Reusing existing code saves time.

Disadvantages of Inheritance:

- The larger the inheritance model gets, the “wider” the mapped table gets, in that for every field in the entire inheritance hierarchy, a column must exist in the mapped table.

Private, public, protected:

- **Public:**

Exp: class A

```
{
```

```
Public:
```

```
Int x;
```

```
Private:
```

```
Int y;
```

```
};
```

```
Class B : public A
```

```
{
```

```
Private:
```

```
Int p;
```

```
}
```

Private, Public & protected

Base class	Derived class visibility		
Visibility	Public derivation	Private derivation	Protected
Private	Not inherited	Not inherited	Not inherited
Public	Public	Private	protected
protected	protected	private	Protected

Private Inheritance:

```
Exp:class a
{
Private:
Int x;
Public:
Int y; };
Class B : private A
{
Int p; };
```

Protected:

- Exp:

```
Class A : protected B
```

```
{
```

```
Private : int X;
```

```
};
```

Abstract class:

An abstract class is one that is not used to create objects. an abstract class is used only for base class.