☆ Public, Protected and Private Inheritance in C++ :-

In C++ inheritance, we can derive a child class from the base class in different access modes

for example :-

```
class Base {
    // some code
};
class Derived : public Base {
    // some code
};
```

Notice the keyword public in the code

class Derived : public Base.

This means that we have created a derived class from the base class in public mode Alternatively, we can also derive classes in protected or private modes.

These 3 keywords {public, protected and private} are known as access specifiers in C++ inheritance.

public, protected and private inheritance have the following features.

- public inheritance makes public members of the base class public in the derived class, and the protected members of the base class remain protected in the derived class

- protected inheritance makes the public and protected members of the base class protected in the derived class.

- private inheritance makes the public and protected members of the base class private in the derived class.

Note :- Private members of the base class are inaccessible to derived class.

Page No.:
Date:

eg:-

```
class Base {
    public:
        int x;
    protected:
        int y;
    private:
        int z;
};

class PublicDerived : public Base {
    // x is public
    // y is protected
    // z is not accessible from PublicDerived.
};

class ProtectedDerived : protected Base {
    // x is protected.
    // y is protected.
    // z is not accessible from protectedDerived
};

class privateDerived : private Base {
    // x is private.
    // y is private.
    // z is not accessible from PrivateDerived
};
```

Example :- C++ public Inheritance.

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

class Base{
  private:
    int pvt = 1;
  protected:
    int prot = 2;
  public:
    int pub = 3;

    // function to access private member.
    int getPVT(){
      return pvt;
    }
};

class PublicDerived : public Base{
  public:
    // function to access protected member
    // from Base.
    int getProt(){
      return prot;
    }
};

int main(){
  PublicDerived object1;
```

```
cout << "Private = " << object1.getPVT() << endl;
cout << "Protected = " << object1.getProt() << endl;
cout << "public = " << object1.pub << endl;

return 0;
}
```

O/P :- Private = 1
Protected = 2
~~protected~~
public = 3.

Here, we have derived PublicDerived from Base in public mode. as a result, in PublicDerived :

- prot is inherited as protected.
- pub and getPVT() are inherited as public.
- pvt is inaccessible since it is prive in base.

Since private and protected members are not accessible, we need to create public functions getPVT() and getProt() to access them.

Accessiblity in Public inheritance.

Accessibility	private member.	protected member	public member
Base class	Yes	yes	yes
Derived class:	No	yes	yes

Example 2 :- C++ protected Inheritance.

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

class Base {
  private:
    int pvt = 1;
  protected:
    int prot = 2;
  public:
    int pub = 3;
    // function to access private member.
    int getPVT() {
      return pvt;
    }
};

class ProtectedDerived : protected Base {
  public:
    // function to access protected member
    // from Base.
    int getProt() {
      return prot;
    }
```

```
    // function to access public member from
    // Base.
    int getPub(){
        return pub;
    }
};

int main(){
    protectedDerived object1;
    cout<<"private can not be accessed"<<endl;
    cout<<"protected = "<<object1.getProt()<<endl;
    cout<<"public = "<<object1.getPub()<<endl;

    return 0;
}
```

O/P:- private cannot be accessed.
protected = 2.
public = 3.

Here, we have derived ProtectedDerived from Base in protected mode. As a result, in ProtectedDerived.

- prot, pub and getPVT() are inherited as protected.

- pvt is inaccessible since it is private in Base.

Date:

As we know, protected members cannot be accessed directly.
As a result, we cannot use getPVT() from ProtectedDerived. that is why also why we need to create the getPub() function in ProtectedDerived in order to access the pub variable.

Accessibility in protected inheritance.

Accessibility	Private members	Protected members	public members
Base class	Yes	Yes	Yes.
Derived class	No	Yes	Yes (inherited as protected variable

Example 3:- C++ private inheritance.

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

class Base {
   private:
      int pvt = 1;
   protected:
      int prot = 2;
   public:
      int pub = 3;
```

```
    //function to access private member.
    int getPVT(){
        return pvt;
    }
};

class PrivateDerived: private Base{
    public:
        //function to access protected member
        //from Base
        int getProt(){
            return prot;
        }

        //function to access private member.
        int getPub(){
            return pub;
        }
};

int main(){
    PrivateDerived object1;
    cout<<"private cannot be accessed"<<endl;
    cout<<"protected = "<<object1.getProt()<<endl;
    cout<<"Public = "<<object1.getPub()<<endl;
    return 0;
}
```

Date: youva

O/P :- Private cannot be accessed.
protected = 2.
public = 3.

Here, we have derived PrivateDerived from Base in private mode. As a result, in PrivateDerived

- prot, pub and getPVT() are inherited as private.
- pvt is inaccessible since it is private in Base

As we know, private members cannot be accessed directly.

As a result, we cannot use getPVT() from PrivateDerived. that is also why we need to create the getPub() function in PrivateDerived in order to access the pub variable.

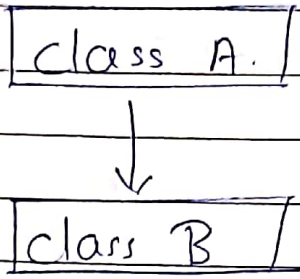
Accessiblity in private ~~member~~ inheritance.

Accessibility	Private member	Protected member	Public member.
Base class	Yes	Yes	Yes.
Derived class	No	yes (inherited as private variable)	yes (inherited as private variable).

Page No.:
Date:

* Single inheritance :- if a single class is derived from one base class then it is called single inheritance.

In C++ single inheritance ~~is~~ based and derived class exhibit one to one relation.



syntax :-
class subclass_name : access_mode base_class.

Example :- previous inheritance example.