1. **Code for Single Inheritence**

**#include <iostream>**

**using namespace std;**

class Account { // Base class or Parent class or SuperClass

public:

float salary = 60000;

};

class Programmer: public Account{ // Derived class or Child class or SubClass

public:

float bonus = 5000;

};

int main() // int main ( void )

{

Programmer p1;

cout<<"Salary:"<<p1.salary<<endl;

cout<<"Bonus:"<<p1.bonus<<endl;

// std::cout<<"Hello World";

return 0;

}

1. **Multilevel Inheritence**

class Animal{

public:

void eat(){

cout<<"Eating..."<<endl;

}

};

class Dog: public Animal

{

public:

void bark(){

cout<<"Barking..."<<endl;

}

};

class BabyDog: public Dog{

publc:

void weep(){

cout<<"Weeping..."<<endl;

}

};

int main(){

BabyDog d1;

d1.eat();

d1.bark();

d1.weep();

return 0;

}

1. **Multiple Inheritence**

class A {

protected:

int a;

public:

void get\_a(int n)

{

a = n;

}

};

class B

{

protected:

int b;

public:

void get\_b(int n){

b = n;

}

};

class C:public A,public B{

public:

void display()

{

cout<<" The value of a is:"<<a<<endl;

cout<<" The value of b is:"<<b<<endl;

cout<<" The Addition of a and b is:"<<a+b;

}

};

int main(){

C c;

c.get\_a(10);

c.get\_b(20);

c.display();

return 0;

}

1. **Multiple Inheritence**

class A

{

protected:

int a;

public:

void get\_a()

{

cout << "Enter the value of 'a':" << endl;

cin >> a;

}

};

class B: public A {

protected:

int b;

public:

void get\_b()

{

cout<<"Enter the value of 'b':"<<endl;

cin>>b;

}

};

class C {

protected:

int c;

public:

void get\_c()

{

cout<<"Enter the value of 'c':"<<endl;

cin>>c;

}

};

class D: public B,public C {

protected:

int d;

public:

void mul()

{

get\_a();

get\_b();

get\_c();

cout << "The Multiplication of a.b.c is:"<<a\*b\*c<<endl;

}

};

int main()

{

D d;

d.mul();

return 0;

}

1. **Hybrid Inheritence**

class A {

public:

void get\_a ()

{

cout<<" Output of class A "<<endl;

}

};

class B : public A {

public:

void get\_b ()

{

cout<<" Output of class B "<<endl;

}

};

class C : public A {

public:

void get\_c ()

{

cout<<" Output of class C "<<endl;

}

};

int main(){

B b;

C c;

b.get\_b();

c.get\_c();

return 0;

}