//Single level inheritance

output

Salary: 60000

Bonus: 5000

#include <iostream>

using namespace std;

class Account {

public:

float salary = 60000;

};

class Programmer: public Account {

public:

float bonus = 5000;

};

int main(void) {

Programmer p1;

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

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

return 0;

}

2-Multiple inheritance

Output-

Eating…

Barking..

#include <iostream>

using namespace std;

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()

{

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

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

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

}

};

int main()

{

C c;

c.get\_a(10);

c.get\_b(20);

c.display();

return 0;

The value of a is : 10

The value of b is : 20

Addition of a and b is : 30

}

3Multi-level Inheritance

Eating...

Barking...

Weeping...

#include <iostream>

using namespace std;

class Animal {

public:

void eat() {

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

}

};

class Dog: public Animal

{

public:

void bark(){

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

}

};

class BabyDog: public Dog

{

public:

void weep() {

cout<<"Weeping...";

}

};

int main(void) {

BabyDog d1;

d1.eat();

d1.bark();

d1.weep();

return 0;

}

#include <iostream>

using namespace std;

class A

{

protected:

int a;

public:

void get\_a()

{

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

cin>>a;

}

};

class B : public A

{

protected:

int b;

public:

void get\_b()

{

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

cin>>b;

}

};

class C

{

protected:

int c;

public:

void get\_c()

{

std::cout << "Enter the value of c is : " << std::endl;

cin>>c;

}

};

class D : public B, public C

{

protected:

int d;

public:

void mul()

{

get\_a();

get\_b();

get\_c();

std::cout << "Multiplication of a,b,c is : " <<a\*b\*c<< std::endl;

}

};

int main()

{

D d;

d.mul();

return 0;

}

5-hierarchiral inheritance

#include <iostream>

using namespace std;

class Shape // Declaration of base class.

{

public:

int a;

int b;

void get\_data(int n,int m)

{

a= n;

b = m;

}

};

class Rectangle : public Shape // inheriting Shape class

{

public:

int rect\_area()

{

int result = a\*b;

return result;

}

};

class Triangle : public Shape // inheriting Shape class

{

public:

int triangle\_area()

{

float result = 0.5\*a\*b;

return result;

}

};

int main()

{

Rectangle r;

Triangle t;

int length,breadth,base,height;

std::cout << "Enter the length and breadth of a rectangle: " << std::endl;

cin>>length>>breadth;

r.get\_data(length,breadth);

int m = r.rect\_area();

std::cout << "Area of the rectangle is : " <<m<< std::endl;

std::cout << "Enter the base and height of the triangle: " << std::endl;

cin>>base>>height;

t.get\_data(base,height);

float n = t.triangle\_area();

std::cout <<"Area of the triangle is : " << n<<std::endl;

return 0;

}