Write a program in C++ to demonstrate the following:

* Multiple inheritance: Create (at least 3) classes of your own choice to demonstrate this concept
* Default & Parameterized Constructors in all classes
* Usage of Friend function for any one class
* Overload any operator to work with objects of classes

#include <iostream>

using namespace std;

class parent1

{

protected:

float data1;

public:

parent1()

{

data1 = 0.0;

}

parent1(float data1)

{

this->data1 = data1;

}

void showData()

{

cout << data1 << endl;

}

};

class parent2

{

protected:

float data2;

public:

parent2()

{

data2 = 0.0;

}

parent2(float data2)

{

this->data2 = data2;

}

void showData()

{

cout << data2 << endl;

}

};

class derived : public parent1, public parent2

{

protected:

float data3;

public:

derived()

{

parent1();

parent2();

data3 = 0.0;

}

derived(float d1, float d2, float d3)

{

data1=d1;

data2=d2;

this->data3 = d3;

}

void showData()

{

parent1::showData();

parent2::showData();

cout<<this->data3<<endl;

}

derived operator ++(int)

{

return derived(++data1,++data2,++data3);

}

friend void square(derived D1);

};

void square(derived D1)

{

cout<<"Square Of "<<D1.data1<<" = "<<D1.data1\*D1.data1<<endl;

cout<<"Square Of "<<D1.data2<<" = "<<D1.data2\*D1.data2<<endl;

cout<<"Square Of "<<D1.data3<<" = "<<D1.data3\*D1.data3<<endl;

}

int main()

{

cout << "Rayyan\t 11217702020\n -------------\n";

derived d1(1.1, 2.2, 3.3);

derived d2;

cout<<"Using Friend Function To Calculate Square"<<endl;

square(d1);

cout<<"Using Post Increment Operator\n"<<endl;

cout<<"d1 Before Increment"<<endl;

d1.showData();

cout<<"d1 After Increment"<<endl;

d2=d1++;

d1.showData();

cout<<"d2 After Increment And Assignment"<<endl;

d2.showData();

return 0;

}

