1. Static Data

Program :

#include<iostream>

using namespace std;

class A {

public:

int a;

static int b;

void increment()

{

a = a+10;

b = b+10;

cout << a << " " << b << endl;

}

};

int A::b=0;

int main()

{

A A1,A2;

A1.a = 20;

A2.a = 30;

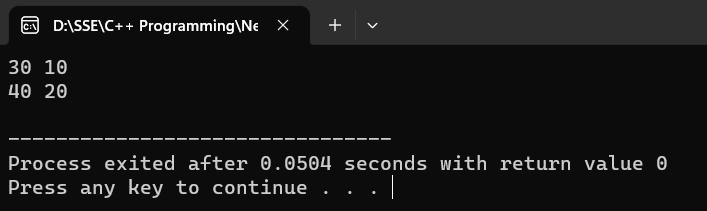
A1.increment();

A2.increment();

return 0;

}

Output :



1. Static Member Function :

Program :

#include<iostream>

using namespace std;

class A {

public:

static int a;

static int b;

void increment()

{

a = a+10;

b = b+10;

cout << a << " " << b << endl;

}

static void decrement()

{

a = a-10;

b = b-10;

}

};

int A::b=0;

int A::a=20;

int main()

{

A A1,A2;

A1.a = 20;

A2.a = 30;

A1.increment();

A2.increment();

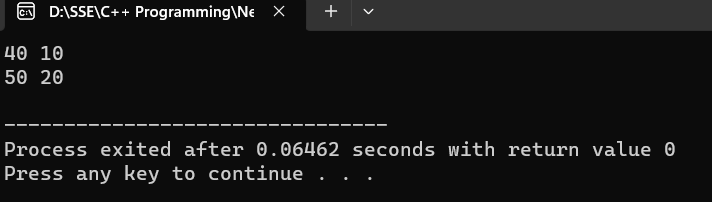
A1.decrement();

A2.decrement();

return 0;

}

Output :



1. Function with default arguments

Program :

#include<iostream>

using namespace std;

class A {

public:

int sum(int a,int b,int c,int d=30,int e=50)

{

return a+b+c+d+e;

}

};

int main()

{

A A1;

cout << A1.sum(10,20,30,100,200) << endl;

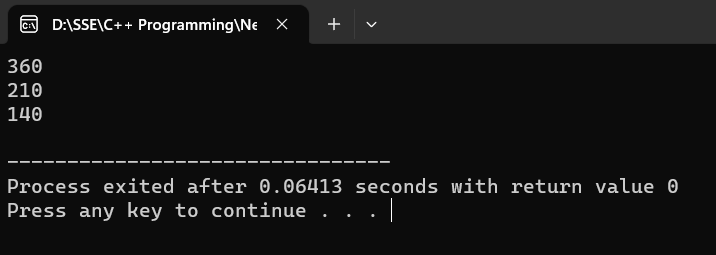
cout << A1.sum(10,20,30,100) << endl;

cout << A1.sum(10,20,30) << endl;

return 0;

}

Output :



1. Virtual based Class

Program :

#include<iostream>

using namespace std;

class A {

public:

void disp()

{

cout << "I'm a virtual based class." << endl;

}

};

class B:virtual public A {

public:

void disp1()

{

cout << "Display1." << endl;

}

};

class C:virtual public A {

public:

void disp2()

{

cout << "Display2." << endl;

}

};

class D: public B,public C {

public:

void disp3()

{

cout << "Display3." << endl;

}

};

int main()

{

D D1;

D1.disp();

return 0;

}

Output :

