**Virtual base class**

#include<iostream>

using namespace std;

class A

{

public:

void display()

{

cout<<"I am a Student:";

}

};

class B:virtual public A

{

public:

void display1()

{

cout<<"bhanu:";

}

};

class C:virtual public A

{

public:

void display2()

{

cout<<"reddy:";

}

};

class D:public B,public C

{

public:

void display3()

{

cout<<"vardhan:";

}

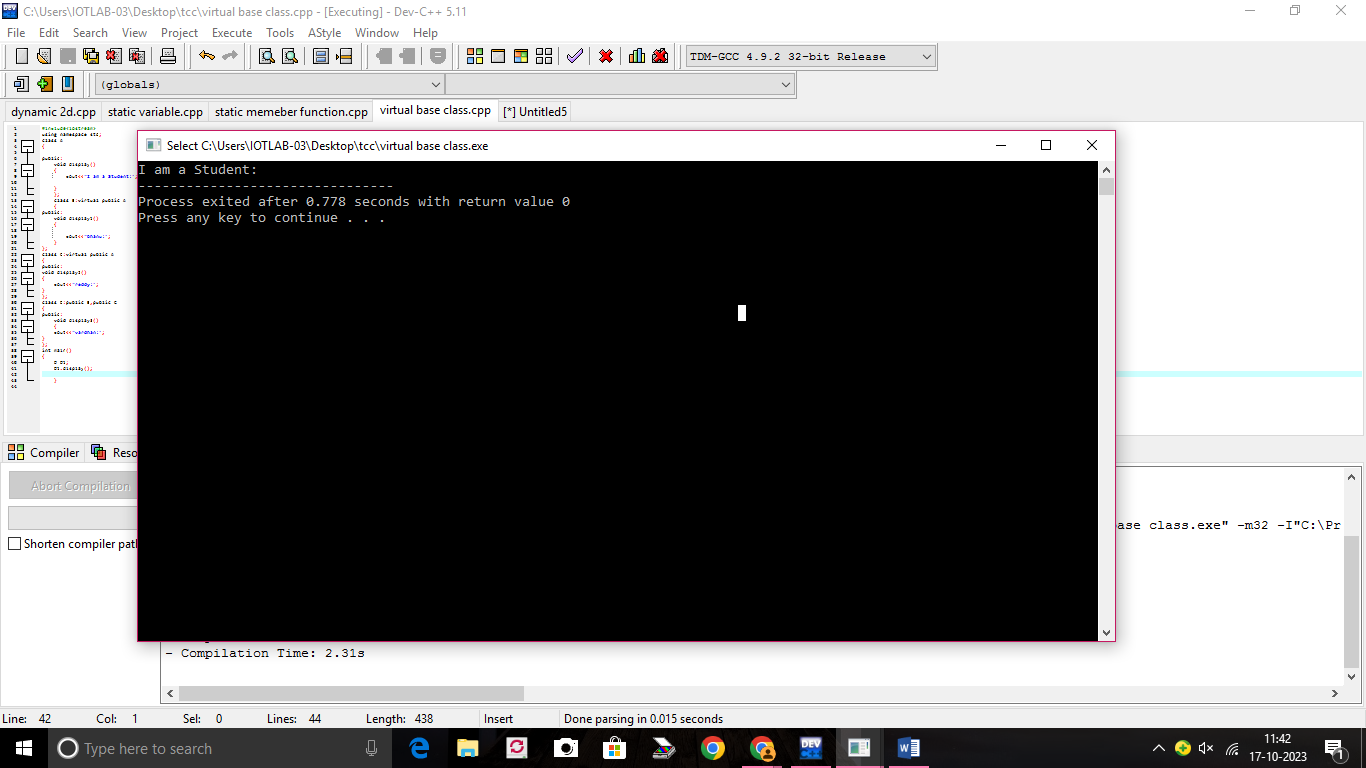
};

int main()

{

D D1;

D1.display();

}

Default argument

#include<iostream>

using namespace std;

class A

{

public:

int sum(int a,int b=40)

{

return a+b;

}

};

int main()

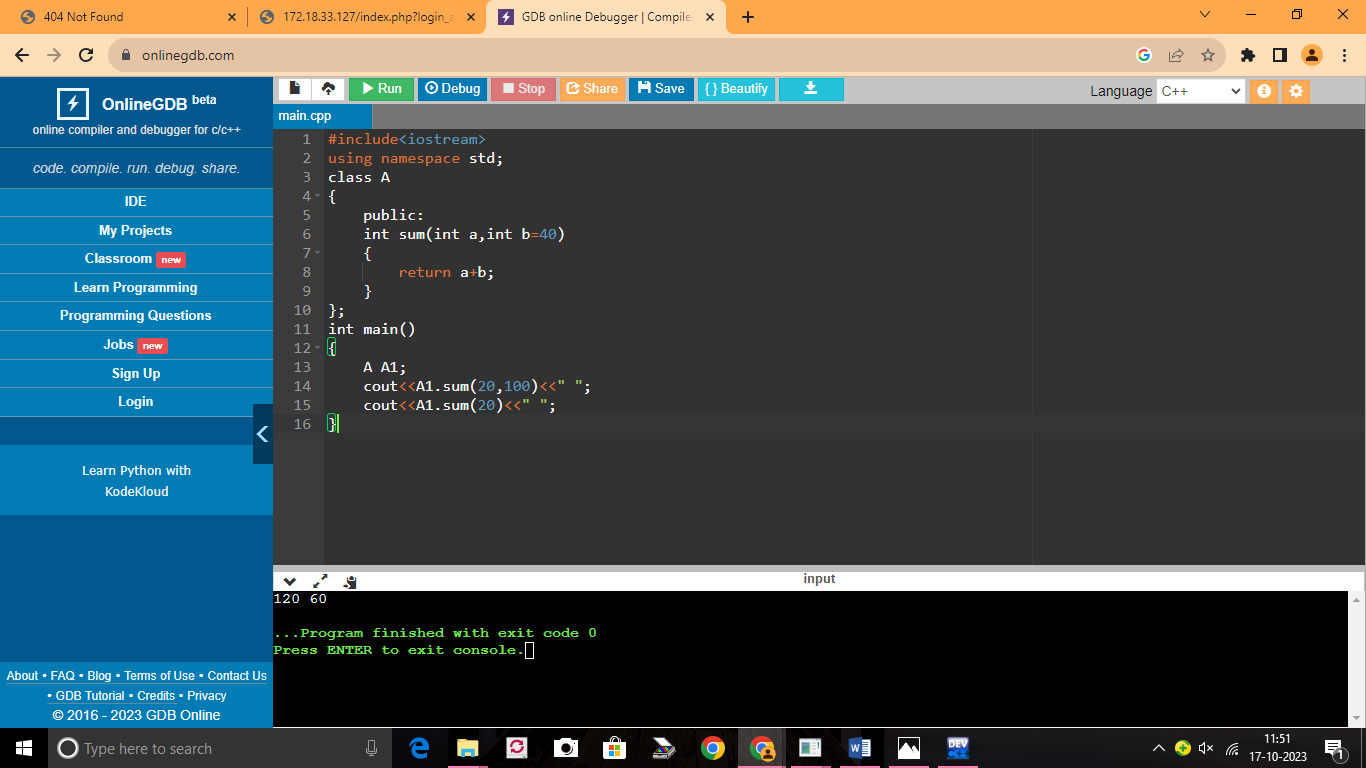
{

A A1;

cout<<A1.sum(20,100)<<" ";

cout<<A1.sum(20)<<" ";

}



Dynamic 2d

#include<iostream>

using namespace std;

int main()

{

int rows=3;

int cols=4;

int \*\*a = new int \*[rows];

for(int i=0;i<rows;i++)

{

a[i]=new int[cols];

}

for(int i=0;i<rows;i++)

{

for(int j=0;j<cols;j++)

{

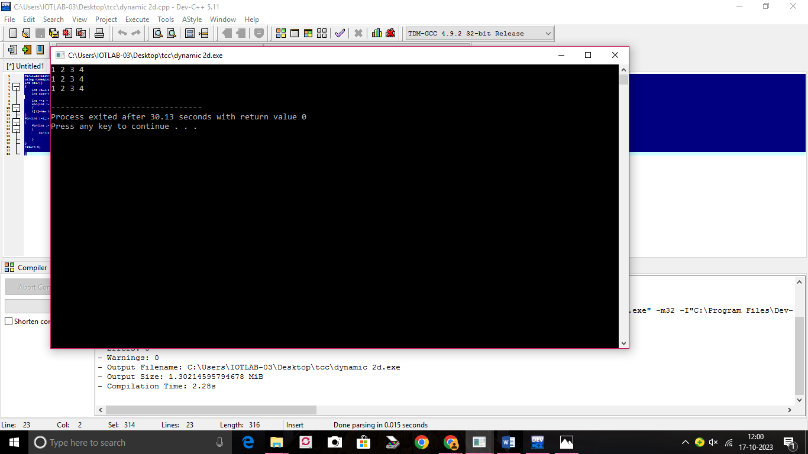
cin>>a[i][j];

}

}

return 0;

}



Static member function

#include <iostream>

using namespace std;

class A {

public:

static int a;

static int b;

static void increment1() {

a = a + 10;

b = b + 10;

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

}

static void decrement1() {

a = a - 10;

b = b - 10;

}

};

int A::b = 0;

int main() {

A A1, A2;

A1.a = 10;

A2.a = 20;

A1.increment1();

A2.increment1();

A1.decrement1();

cout << A1.a << " " <<A2.a<< A::b << " ";

return 0;

}