**EXPERIMENT – 5(A)**

**PROGRAM :** Calculate area of circle, rectangle, triangle using function overloading **.**

**PSEUDOCODE :**

1. Define a macro PI 3.14
2. Make a class ABC.
3. Construct 3 area member functions ,with different default arguments.
4. Area of circle= PI\*r\*r ,rectangle =a\*b
5. Area of Triangle = s=(x+y+z)/2, area = sqrt(s(s-a)(s-b)(s-c))
6. Start Main.
7. Define three objects of ABC type c,r,t.
8. Call parameterized member functions using objects, with different parameters.
9. End Main.

**CODE :**

#include<iostream>

#include<cmath>

#define PI 3.14

using namespace std;

class ABC

{

public :

int area(int r)

{

cout<<"Area of circle is "<<PI\*r\*r<<endl;

return 0;

}

int area (int a,int b)

{

cout<<"Area of the rectangle is : "<<a\*b<<endl;

return 0;

}

int area(int x,int y,int z )

{

float sem = (x+y+z)/2;

int ans = sem\*(sem-x)\*(sem-y)\*(sem-z);

int out = sqrt(ans);

cout <<"The area of the triangle is : "<<out<<endl;

return 0;

}

};

int main()

{ ABC c,r,t;

c.area(2);

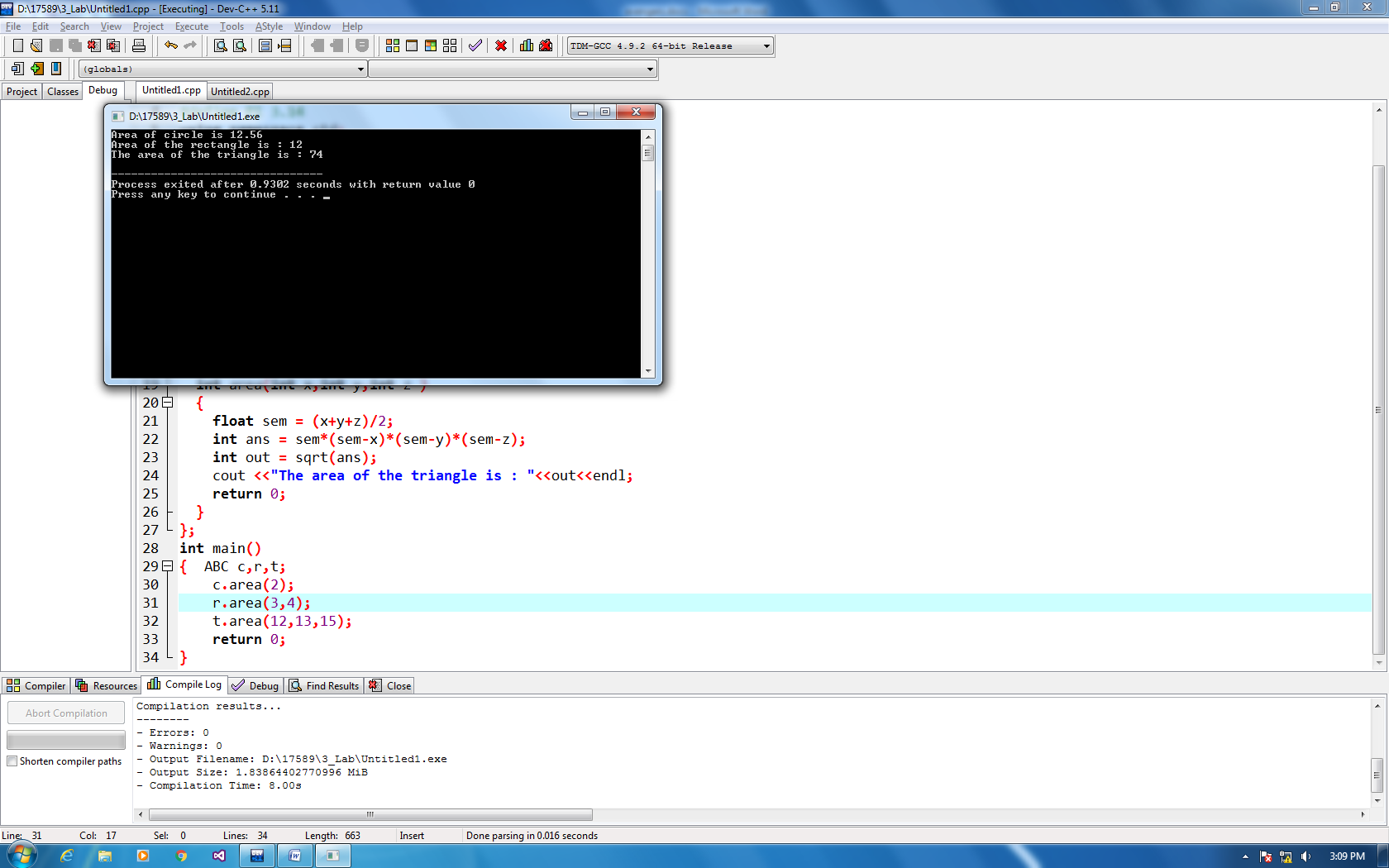
r.area(3,4);

t.area(12,13,15);

return 0;

}

**OUTPUT :**



**EXPERIMENT - 5(B)**

**PROGRAM :** Write a program in C++ to find multiplication of two numbers and cube using inline function.

**PSEUDOCODE :**

1. Define a class A.
2. Define a,b,c in private data members.
3. Define 2 inline functions multiply , cube.
4. Pass 2 arguments in multiply function.
5. Store the arguments in the private data . (a,b)
6. Product = a\*b
7. Pass 1 argument in cube function. (z)
8. Store the arguments in private data . (c)
9. Cube = c\*c\*c
10. In main, make a object a of class A.
11. Call a.multiply (argument ,argument) and a.cube(arguments).
12. End Main.

**CODE :**

#include<bits/stdc++.h>

using namespace std;

class A

{

private :

int a,b,c;

public :

inline int multiply(int x,int y)

{ a=x;

b=y;

cout << "Product of the two numbers is : "<<a\*b<<endl;

return 0;

}

inline int cube (int z )

{

c=z;

cout<<"The cube of the number is : "<<c\*c\*c<<endl;

return 0;

}

};

int main()

{

A a;

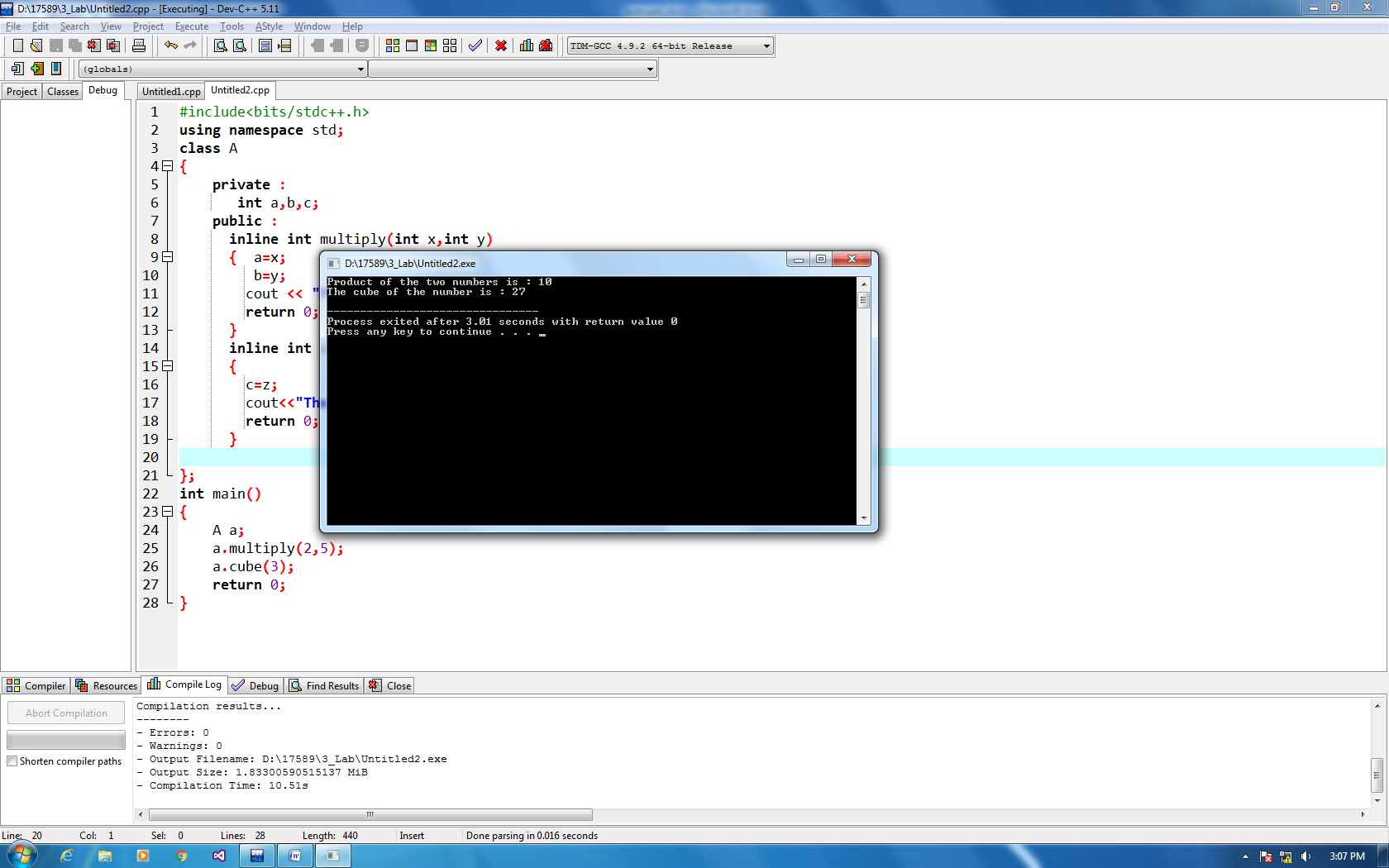
a.multiply(2,5);

a.cube(3);

return 0;

}

**OUTPUT :**



**EXPERIMENT -5(C)  
PROGRAM :** Write a program in C++ to enter a student data and display it using classes.

**PSEUDOCODE :**

**CODE :**

1. Define a class student.
2. Define two data members a character name array and a int type roll no.
3. In public define 2 member functions getdata() and display().
4. In getdata input name and roll no and store in private data members.
5. In display put name and roll no.
6. End Class.
7. In main.
8. Define objects s1,s2 of class student.
9. Get data using s1.getdata() and put data using s1.display().
10. End main.

**#i**nclude<iostream>

using namespace std ;

class student

{ private :

char name[20];

int roll\_no;

public :

void getdata()

{

cout << "Enter your name :"<<endl;

cin>> name ;

cout <<"Enter your roll no : ";

cin>> roll\_no;

}

void display()

{

cout<<"The details entered by the student are: "<< endl;

cout << name<<endl;

cout<< roll\_no<<endl;

}

};

int main()

{

student s1,s2;

s1.getdata();

s2.getdata();

s1.display();

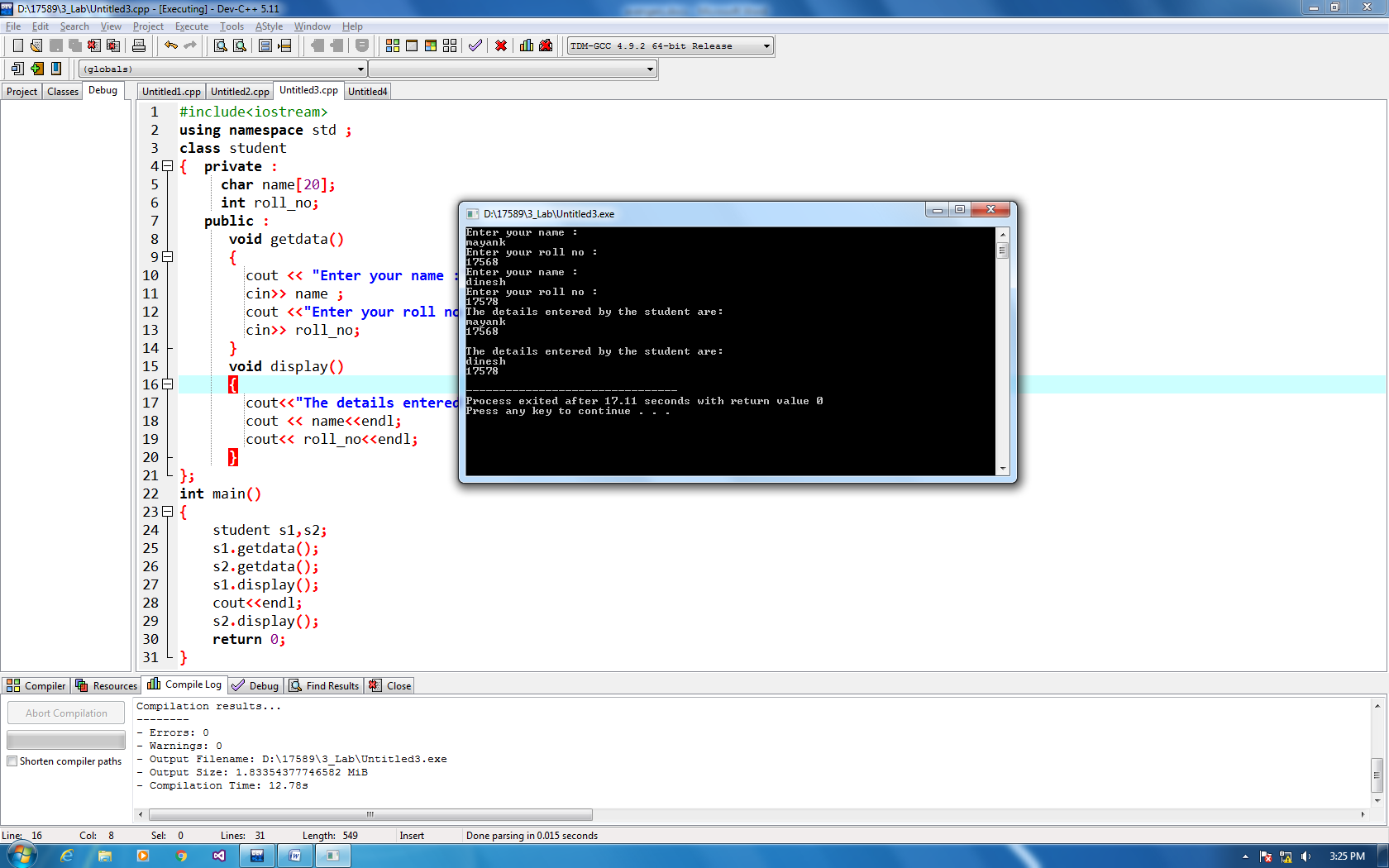
cout<<endl;

s2.display();

return 0;

}

**OUTPUT :**



**EXPERIMENT – 5(D)**

**PROGRAM :** Write a program in C++ to demonstrate the use of friend function in classes .

**PSEUDOCODE :**

1. Declare 2 classes A and B.
2. In class B , declare private data x .
3. In public declare a friend function display() with arguments as classes.
4. Define get function to input data and x = input.
5. Similarly in class B get y = input.
6. Define the friend function. Output the sum as A.y+B.x
7. Start Main
8. Define objects q of class A and w of class B.
9. Get data and display
10. End main.

**CODE :**

#include<iostream>

using namespace std;

class a;

class b

{

private :

int x;

public :

friend void display(class a,class b);

void get(int h)

{

x=h;

}

};

class a

{

private :

int y;

public :

friend void display(class a,class b);

void get(int v)

{

y=v;

}

};

void display(a A ,b B )

{

cout << A.y+B.x;

}

int main()

{ a q;

b w;

q.get(12);

w.get(14);

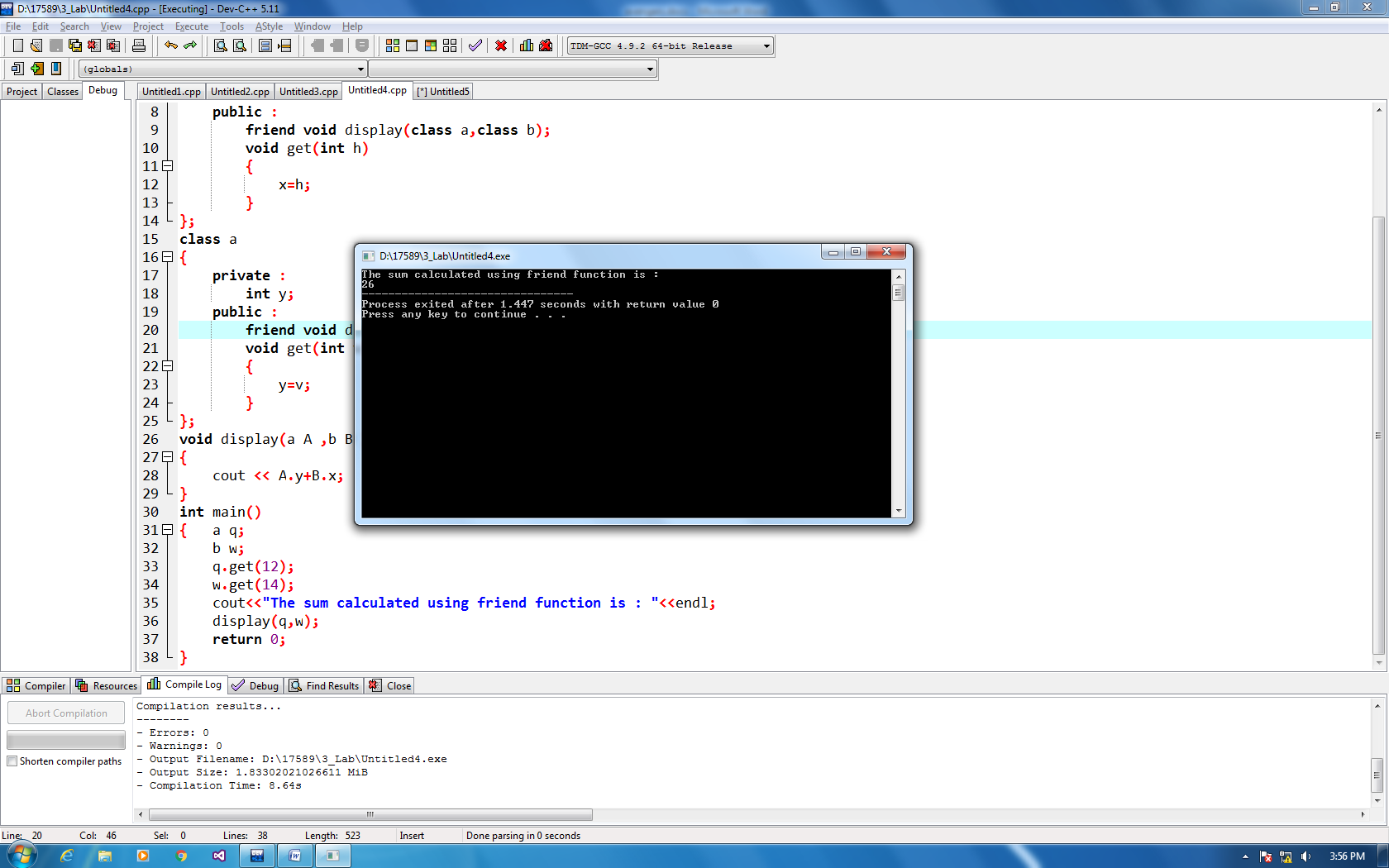
cout<<"The sum calculated using friend function is : "<<endl;

display(q,w);

return 0;

}

**OUTPUT :**



**EXPERIMENT – 5(E)**

**PROGRAM :** Write a program in C++ to calculate factorial of a number using copy constructor.

**PSEUDOCODE :**

1. Define a class fact.
2. Declare 3 data members I,n,f in private .
3. In Public declare a constructor fact with one default argument.
4. Declare a copy constructor with argument as address .
5. Define a call function with no arguments.
6. Set for loop and loop for i<n with increment of 1.
7. Store the results in f.
8. Declare a function display. Output the value of f.
9. Start Main.
10. Input x till where factorial is to be calculated.
11. Call the factorial and pass value of x.
12. Call the copy constructor and pass the default constructor as arguments.
13. Output display of f2.
14. End Main.

**CODE :**

#include<iostream>

using namespace std;

class fact

{

int n, i, f;

public:

fact(int x)

{

n=x;

f=1;

}

fact(fact &x)

{

n=x.n;

f=1;

}

void cal()

{

for(i=1; i<=n; i++)

{

f=f\*i;

}

}

void display()

{

cout<<"Factorial of the number is : "<<f<<endl;

}

};

int main()

{ int x;

cout<<"Enter a number"<<endl;

cin>>x;

fact f1(x);

f1.cal();

f1.display();

fact f2(f1);

f2.cal();

f2.display();

return 0;

}

**OUTPUT :**

