***NAME : Himanshu Dixit***

***ENROLL NO. : 21103262***

***BATCH : B10***

***Software Development Lab – II [15B17CI271]***

***Assignment Sheet***

***Week 3***

***Q1.*** *WAP to print the area and perimeter of a triangle having sides 3, 4, 5 units  creating a class Triangle with a function to print area and perimeter.*

***Solution :***

#include <iostream>

#include <cmath>

using namespace std;

class triangle

{

int a,b,c;

float peri,area;

public:

void display\_peri(int a, int b, int c)

{

peri=a+b+c;

cout<<"Perimeter is : "<<peri<<endl;

}

void display\_area(int a, int b, int c)

{

int s = (a+b+c)/2.0;

area = sqrt(s\*(s-a)\*(s-b)\*(s-c));

cout<<"Area is : "<<area;

}

};

int main()

{

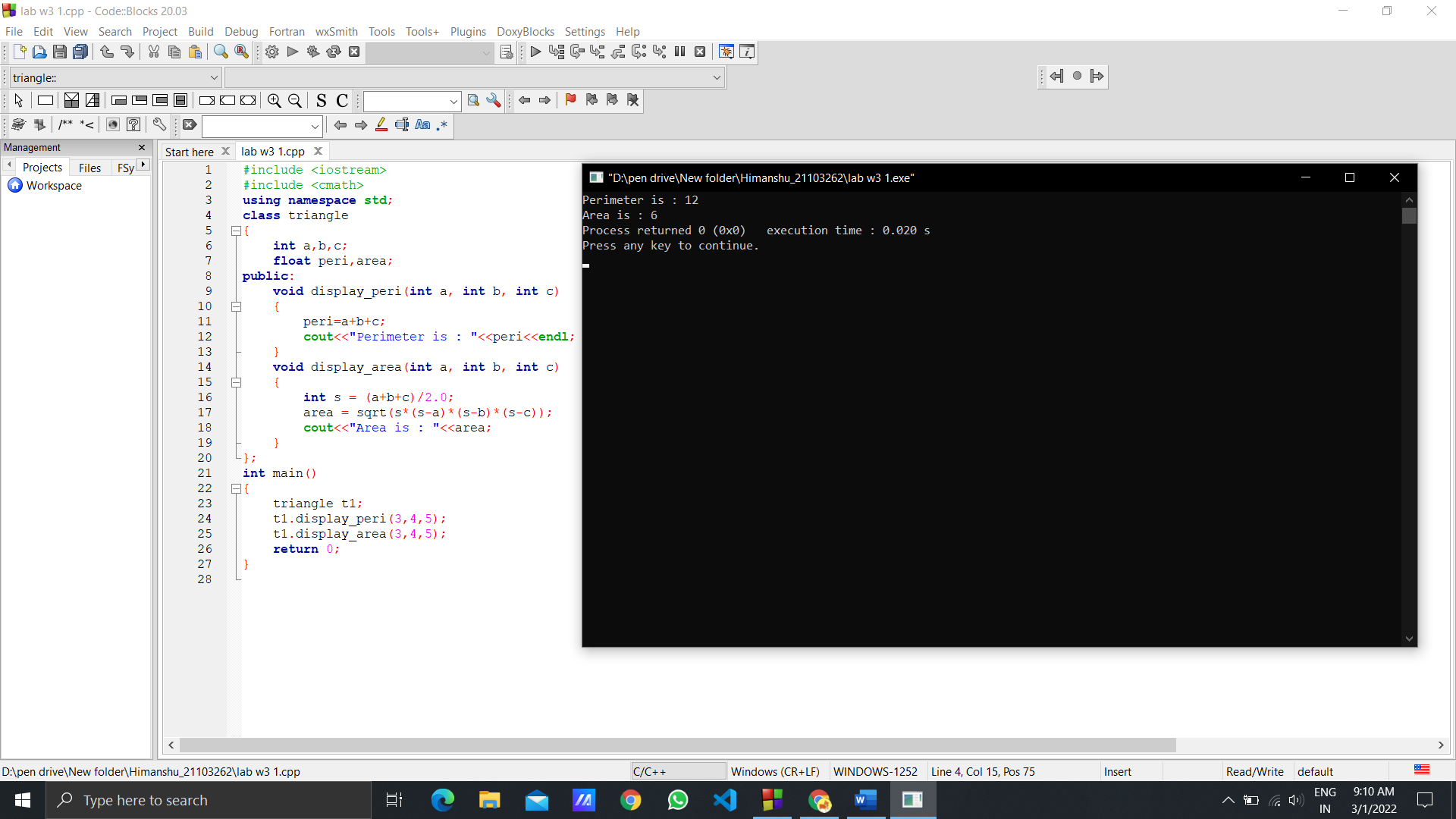
triangle t1;

t1.display\_peri(3,4,5);

t1.display\_area(3,4,5);

return 0;

}



***Q2.*** *Modify Q1 to add a constructor that takes the sides as parameters.*

***Solution :***

#include <iostream>

#include <cmath>

using namespace std;

class triangle

{

int a,b,c;

float peri,area;

public:

triangle(int x, int y, int z)

{

a=x;

b=y;

c=z;

}

void display\_peri()

{

peri=a+b+c;

cout<<"Perimeter is : "<<peri<<endl;

}

void display\_area()

{

int s = (a+b+c)/2.0;

area = sqrt(s\*(s-a)\*(s-b)\*(s-c));

cout<<"Area is : "<<area;

}

};

int main()

{

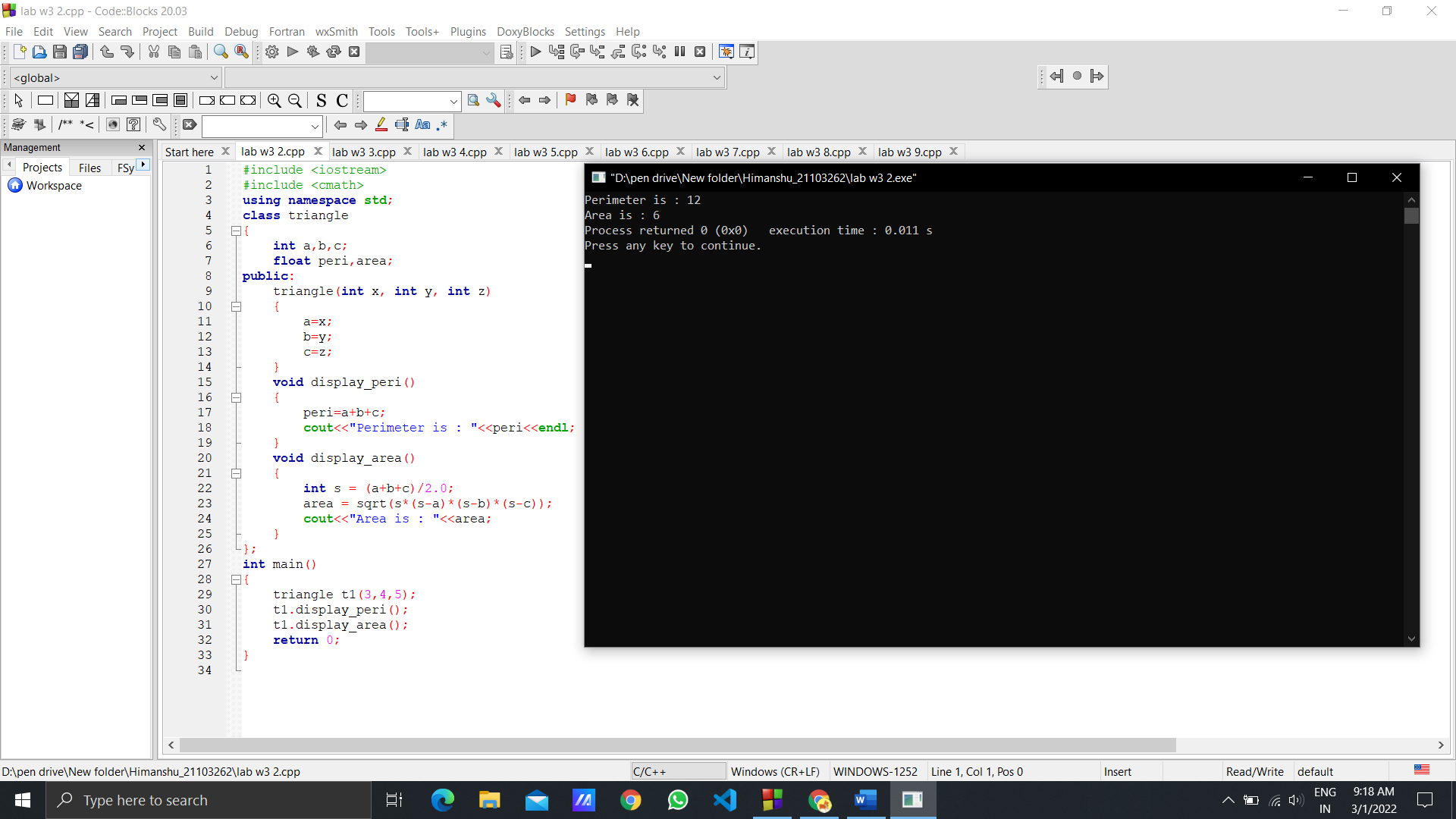
triangle t1(3,4,5);

t1.display\_peri();

t1.display\_area();

return 0;

}



***Q3.*** *Create a class named Complex, with separate functions to get real parts,  complex parts and print Sum, Difference and Product of the complex numbers.*

***Solution :***

#include <iostream>

#include <cmath>

using namespace std;

class complex

{

int r1,r2,c1,c2;

float s,mul,sub;

public:

void real\_part()

{

cout<<"Enter Real part of complex number 1 and 2 : \n";

cin>>r1>>r2;

}

void complex\_part()

{

cout<<"Enter Complex part of complex number 1 and 2 : \n";

cin>>c1>>c2;

}

void sum()

{

s=r1+r2;

cout<<"\nSum is : "<<s;

s=c1+c2;

cout<<" + i"<<s;

}

void diff()

{

sub=r1-r2;

cout<<"\nDifference is : "<<sub;

sub=c1-c2;

if(sub>=0)

cout<<" + i"<<sub;

else

cout<<" - i"<<abs(sub);

}

void product()

{

mul=r1\*r2 - c1\*c2;

cout<<"\nMultiplication is : "<<mul;

mul=r1\*c2 + r2\*c1;

if(mul>=0)

cout<<" + i"<<mul;

else

cout<<" - i"<<abs(mul);

}

};

int main()

{

complex c;

c.real\_part();

c.complex\_part();

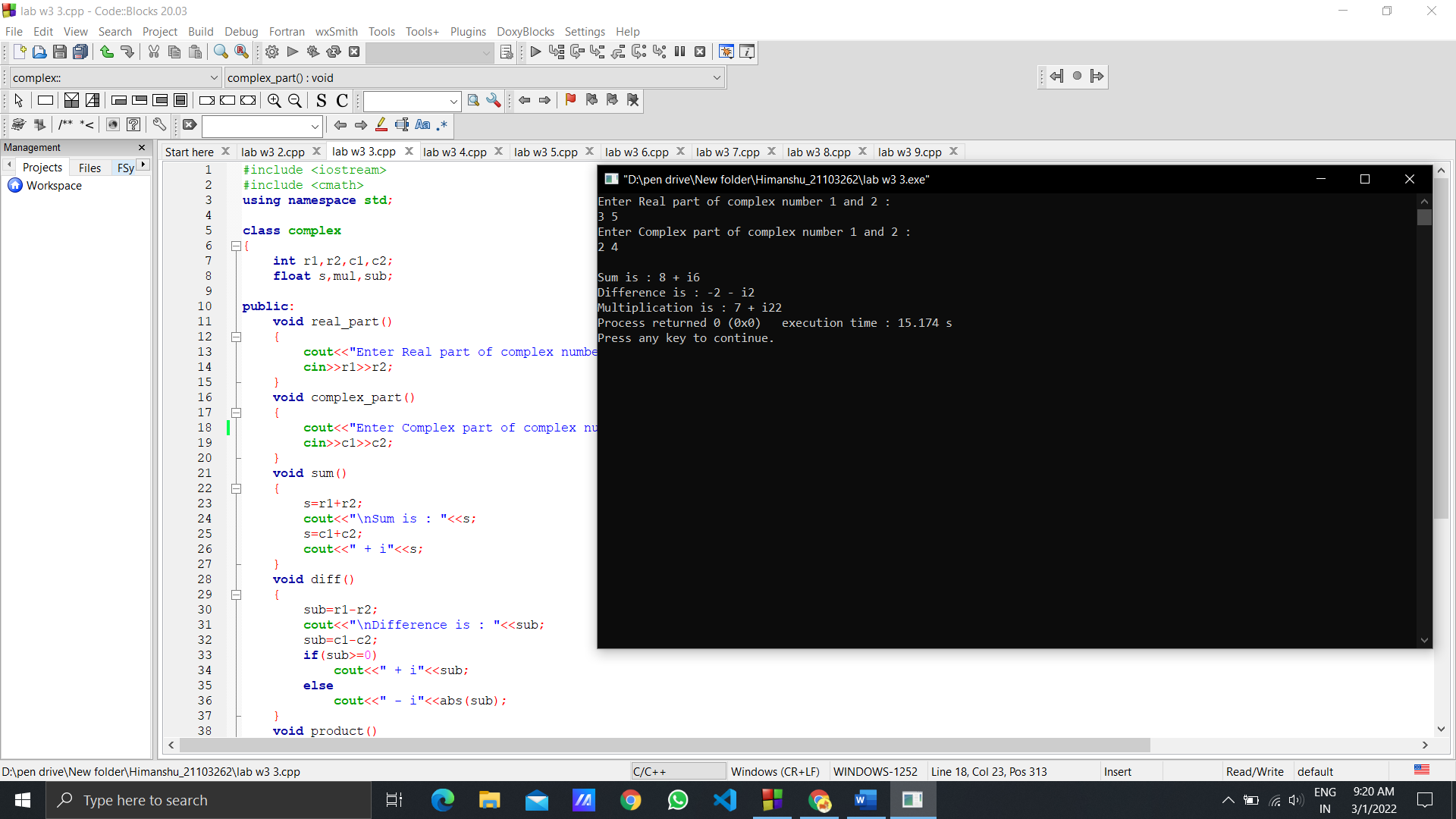
c.sum();

c.diff();

c.product();

return 0;

}



***Q4.*** *WAP in C++ to demonstrate the following:*

***(i)*** *Create a class called First with data member as string studentname, and  member function called printname*

***(ii)*** *Your program should print the studentname by accessing the member  function.*

***Solution :***

#include <iostream>

#include <stdio.h>

using namespace std;

class first

{

char studentname[100];

public:

void printname(const char \*n)

{

cout<<"Student name is : "<<n;

}

};

int main()

{

first f;

char name[100];

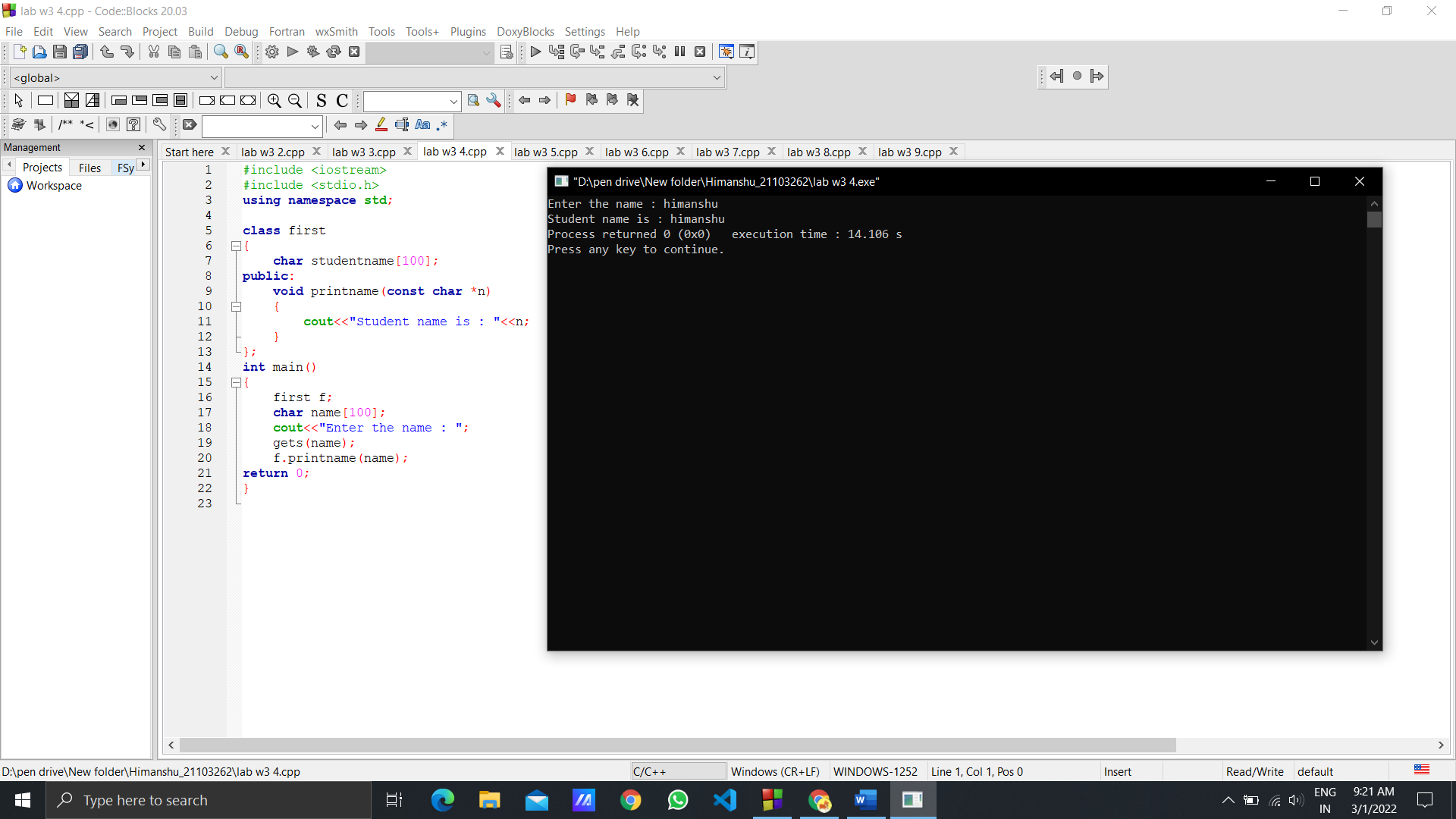
cout<<"Enter the name : ";

gets(name);

f.printname(name);

return 0;

}



***Q5.*** *WAP in C++with the following requirements:*

***(i)*** *Create a class called Second with int type person id*

***(ii)*** *Create a default and parameterized constructors*

***(iii)*** *Create two objects. obj1 will call default constructor to print person id, obj1  will call the parameterized constructor to print person id.*

***Solution :***

#include <iostream>

using namespace std;

class Second

{

int id;

public:

Second()

{

cout<<"Enter the person id : ";

cin>>id;

cout<<"Id is : "<<id;

}

Second(int a)

{

id=a;

cout<<"\nId is : "<<id;

}

};

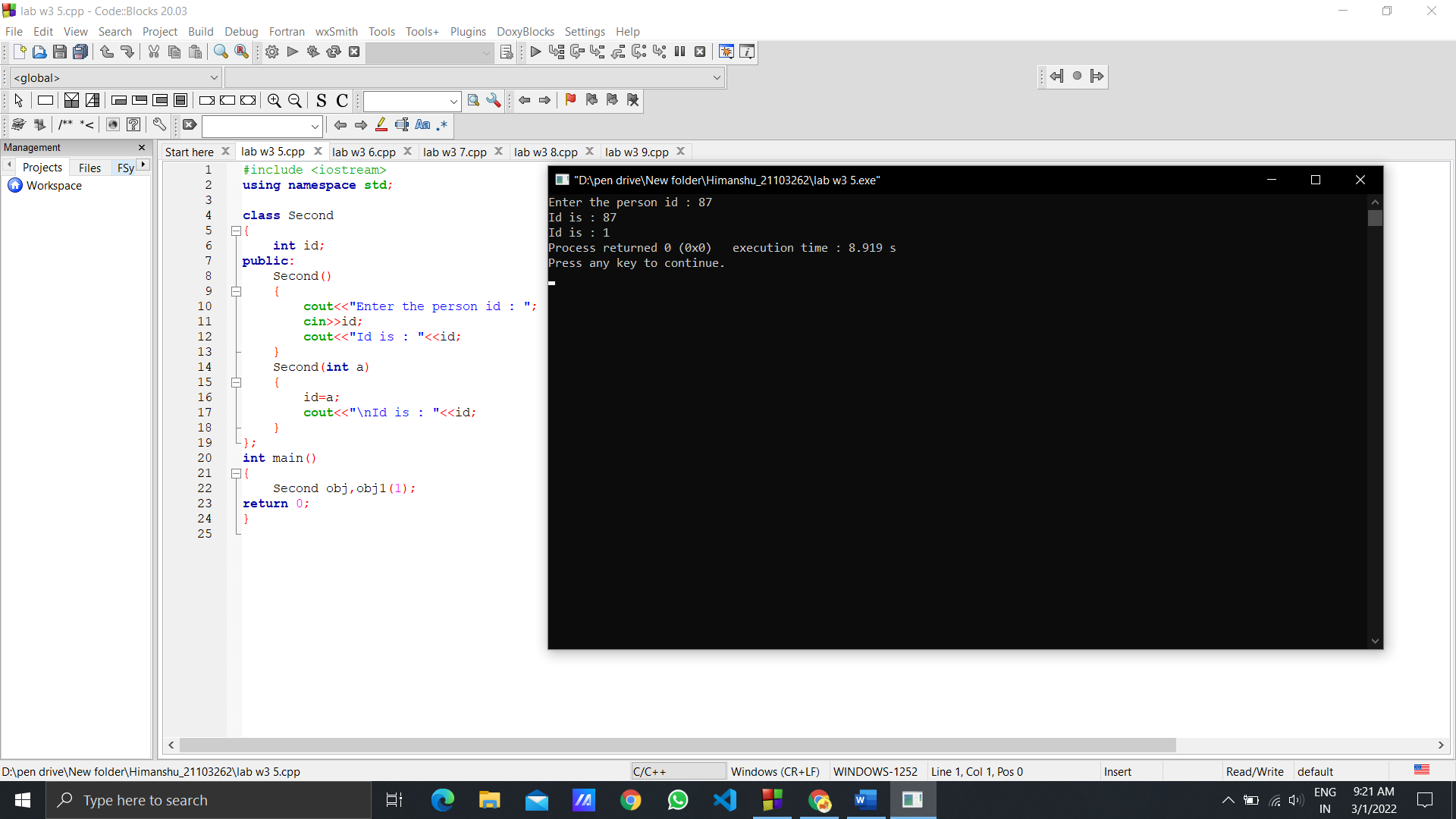
int main()

{

Second obj,obj1(1);

return 0;

}



***Q6.*** *Write a C++ program to print the area of a rectangle by creating a class named  Area having two functions. First named setDim() that takes length and breadth of the  rectangle and the second named getArea() that retunrs the area of the rectangle.*

***Solution :***

#include <iostream>

using namespace std;

class Area

{

int l;

int b;

public:

void setDim()

{

cout<<"Enter the Length and Breadth : ";

cin>>l>>b;

}

int getArea()

{

return l\*b ;

}

};

int main()

{

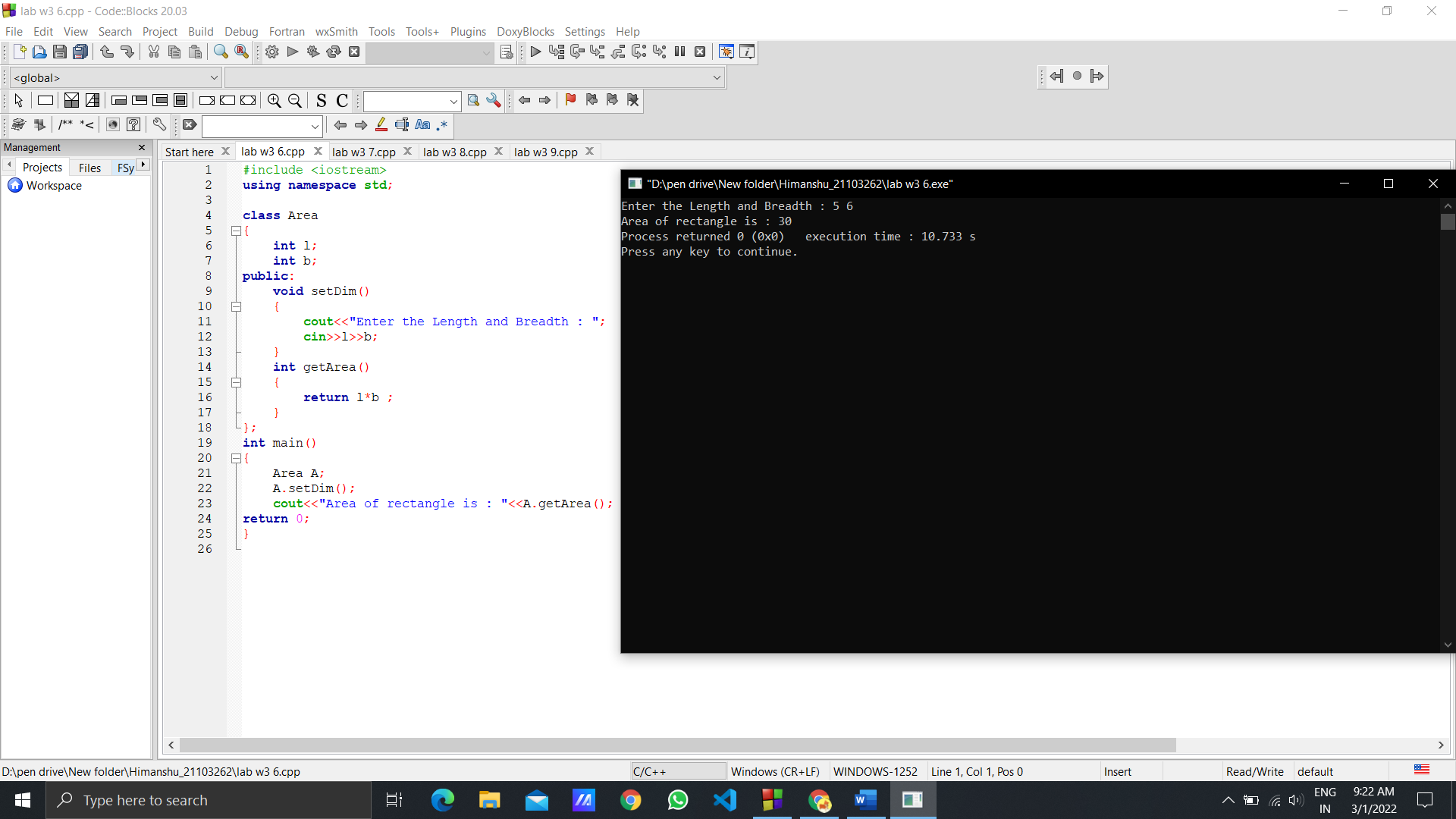
Area A;

A.setDim();

cout<<"Area of rectangle is : "<<A.getArea();

return 0;

}



***Q7.*** *Write a C++ program to calculate the average marks of two students. Create a  class named Student. Create a parameterized constructor that initializes student  marks and a member function calculateAverage() that accepts Student objects as  parameters.*

***Solution :***

#include <iostream>

using namespace std;

class Student

{

int sub1,sub2;

public:

Student(int a,int b)

{

sub1=a;

sub2=b;

}

void calculateAverage(Student S)

{

cout<<"Average is : "<<(S.sub1+S.sub2)/2;

}

};

int main()

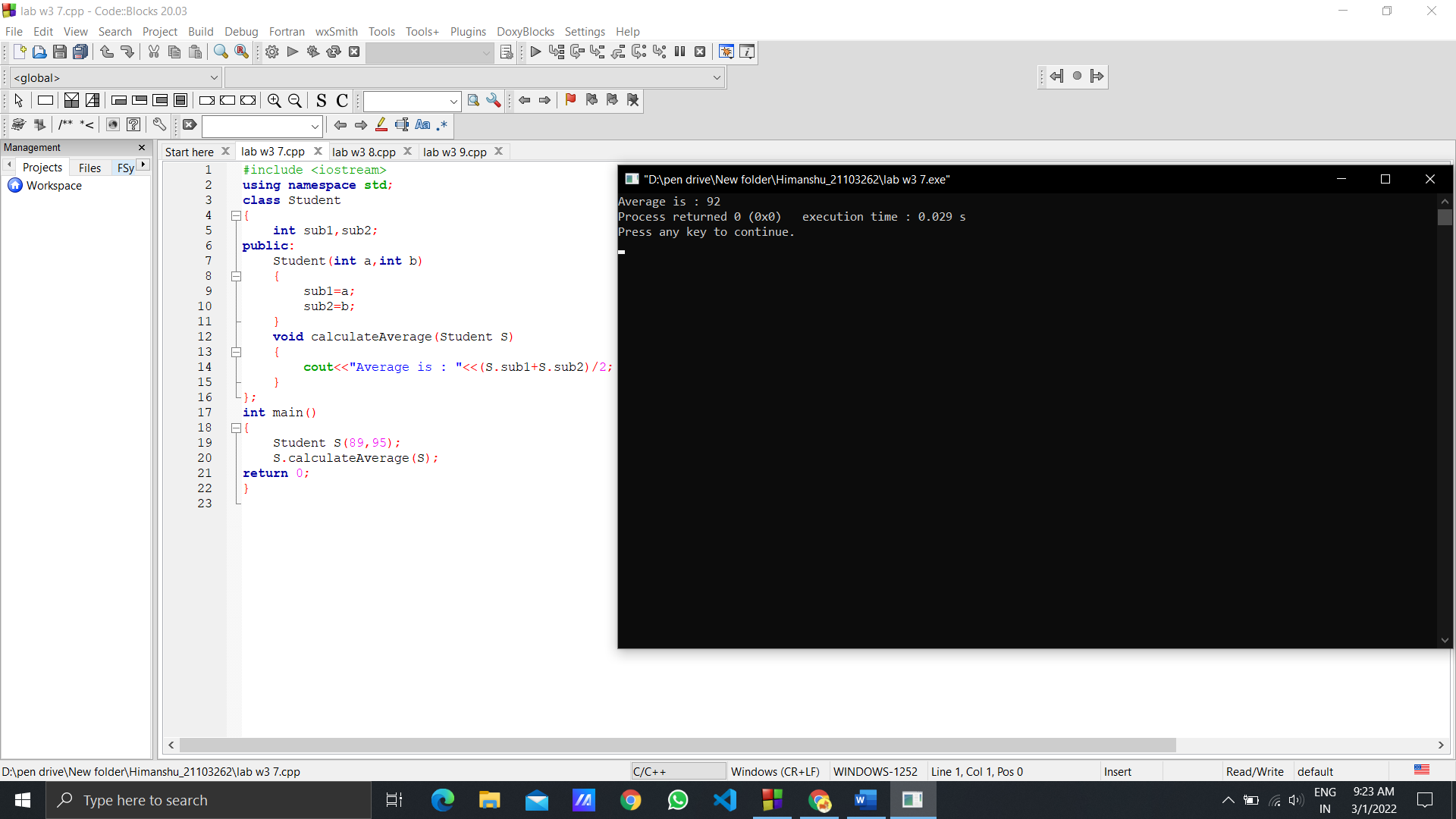
{

Student S(89,95);

S.calculateAverage(S);

return 0;

}



***Q8.*** *Write a C++ program to create a class that can hold private data members such  as – Enrolment number, Name, Branch and CGPA. Include the constructors for  initialising data members as follows,*

***(i)*** *Initialise the enrolment no., name, branch and CGPA*

***(ii)*** *Initialise the enrolment no., name, CGPA (Default branch=’CSE’)*

***(iii)*** *Initialise enrolment no., name, branch and CGPA with default values [1,  “Unknown”, “CSE”, 6]*

*Also include get and set functions for each of the data members. Create objects to  call different constructors and update CGPA value of any of the objects created.  Display the updated CGPA along with the student’s detail on the console.*

***Solution :***

#include <iostream>

using namespace std;

class studentinfo

{

int enrollmentno;

string name;

string branch;

double CGPA;

public:

studentinfo()

{

enrollmentno=1;

name="Unknown";

branch="CSE";

CGPA=6.0;

}

studentinfo(int a,string b,double c)

{

enrollmentno=a;

name=b;

branch="CSE";

CGPA=c;

}

studentinfo(int a,string b,string c,double d)

{

enrollmentno=a;

name=b;

branch=c;

CGPA=d;

}

void setdata()

{

cin>>enrollmentno>>name>>branch>>CGPA;

}

void getdata()

{

cout<<enrollmentno<<" "<<name<<" "<<branch<<" "<<CGPA<<"\n";

}

void update(string b)

{

branch=b;

}

};

int main()

{

studentinfo s;

s.getdata();

studentinfo s1(1,"himanshu",9.2);

s1.update("IT");

s1.getdata();

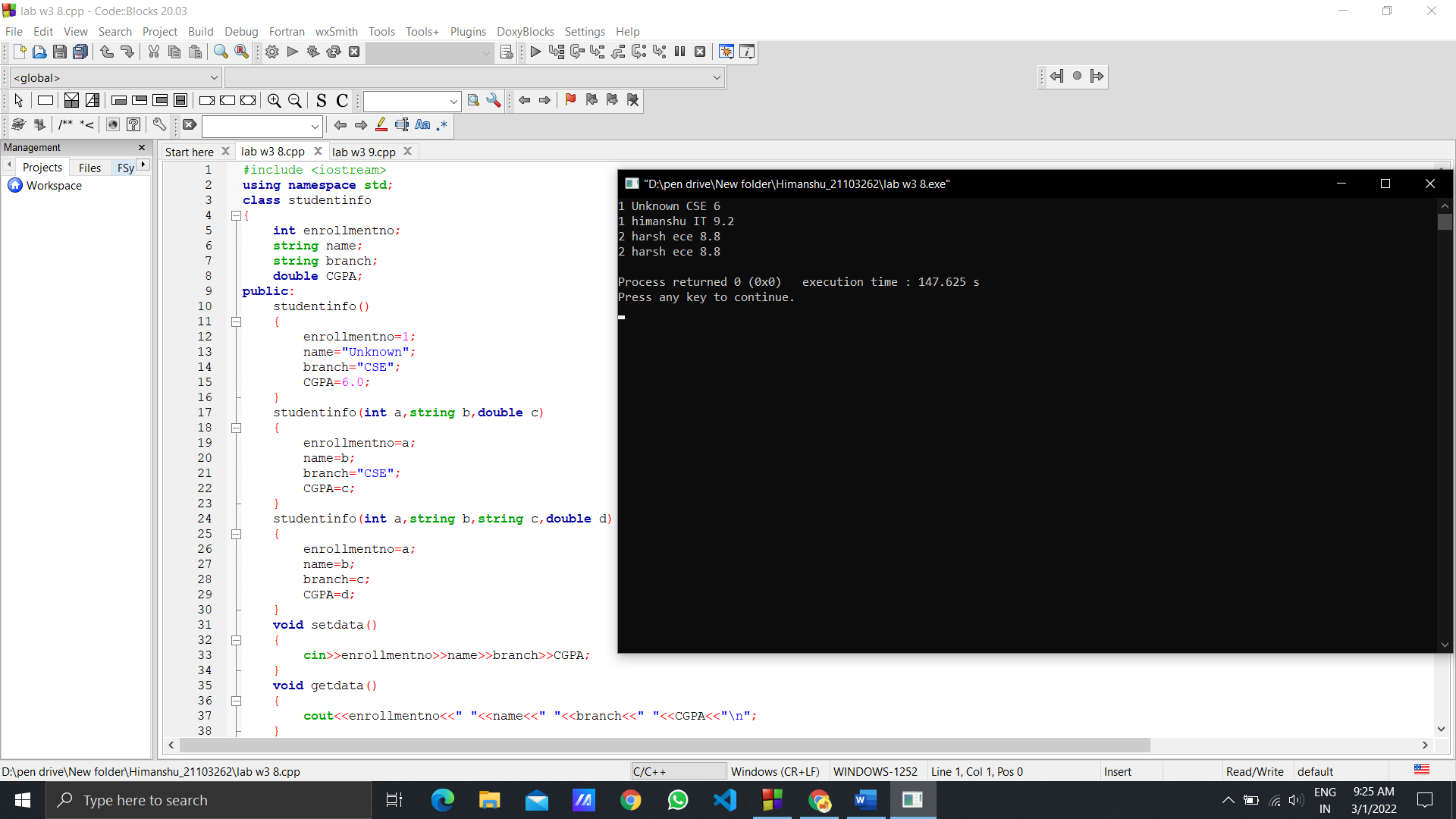
studentinfo s2(1300,"appa","ECE",9);

s2.setdata();

s2.getdata();

return 0;

}



***Q9.*** *Write a C++ program to calculate electricity bill using Class.*

***∙*** *To Calculate Electricity Bill Of Person using Class,first create and call get( )  function to take input details of the customer.*

***∙*** *Create and call a new function i.e calc\_bill( ) to calculate the total bill of the  customer on the behalf of units consumed by the customer .*

***∙*** *Create the put( ) function to print or display customer or person electricity  bill on the screen.*

***Solution :***

#include <iostream>

using namespace std;

class bill

{

string name;

int billno;

int units;

public:

void get(void);

void put(void);

float calc()

{

float total;

if (units <= 200)

{

total = 0;

}

else if (units <= 500)

{

total = (units - 200) \* 3;

}

else if (units <= 1000)

{

total = (units-500)\*4+300\*3;

}

else

{

total = (units - 1000) \* 4 + 500 \* 4 + 300\*3;

}

return total;

}

};

void bill::get()

{

cout<<"enter name , billno , units consumed"<<endl;

cin>>name>>billno>>units;

}

void bill::put()

{

cout<<"your electricity bill is"<<endl;

cout<<calc();

}

int main()

{

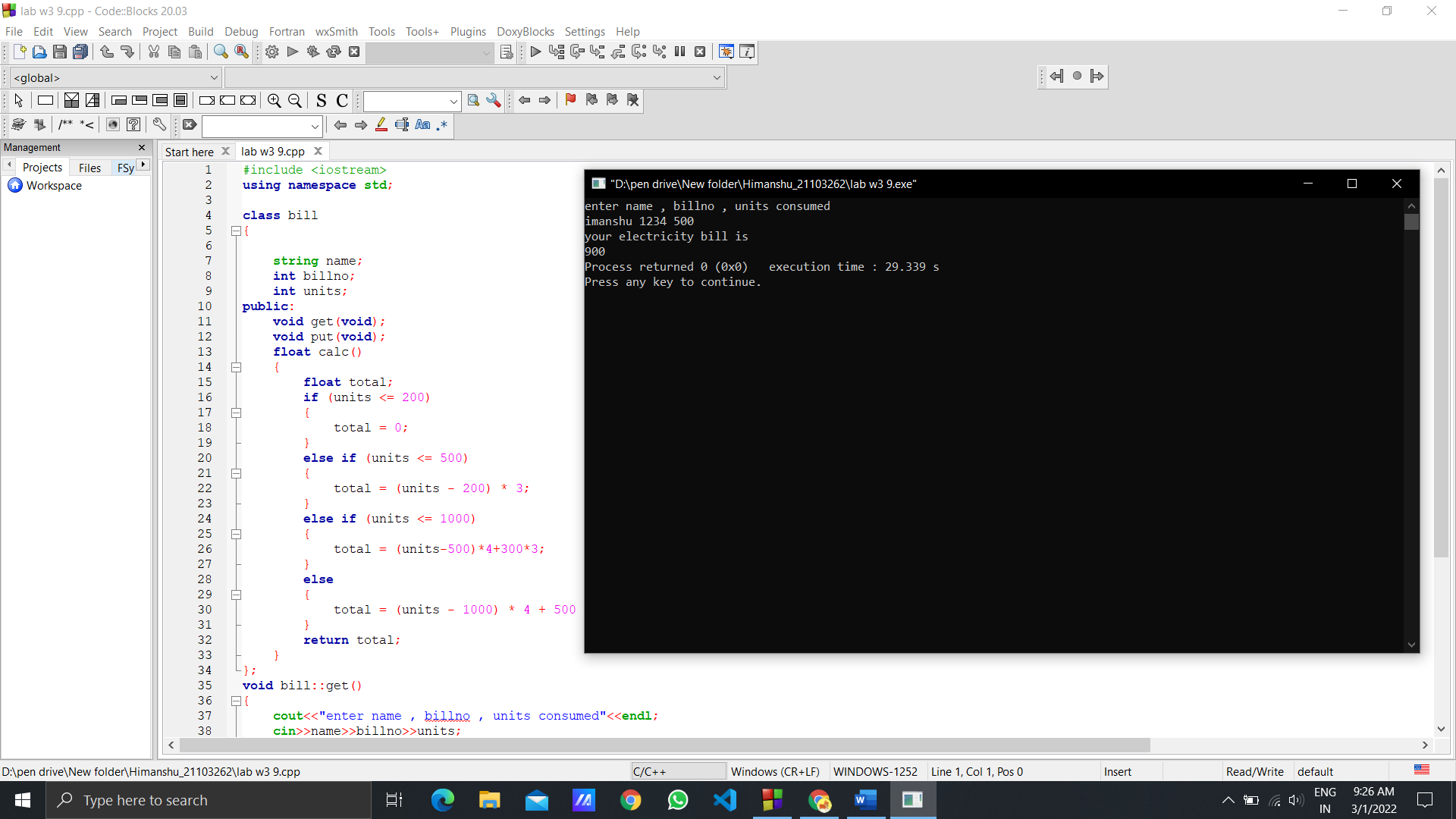
bill b;

b.get();

b.put();

return 0;

}



***Q10.*** *What would be the output of the following program in C++?*

*// C++ program to explain constructors*

*#include <iostream>*

*using namespace std;*

*class Location*

*{*

*private:*

*inta, b;*

*public:*

*Location(int a1, int b1) { a = a1; b = b1; }*

*// Copy constructor*

*Location(const Location &l2) {a = l2.a; b = l2.b; }*

*int getA() { return a; }*

*int getB() { return b; }*

*};*

*int main()*

*{*

*Location l1(10, 15); // Normal constructor is called here*

*Location l2 = l1; // Copy constructor is called here*

*// Let us access values assigned by constructors*

*cout <<"l1.a = "<<l1.getA() <<", l1.b = "<<l1.getB();*

*cout <<"\nl2.a = "<<l2.getA() <<", l2.b = "<<l2.getB();*

*return 0;*

*}*

***Solution :***

*l1.a = 10, l1.b = 15*

*l2.a = 10, l2.b = 15*