Assignment Constructor and friend functiond friend function

1.Answer the questions (i) and (iii) after going through the following class:

class Seminar

{

int time;

public:

Seminar() //Function 1

{

time = 30;

cout &lt;&lt; &quot;Seminar starts now&quot; &lt;&lt; endl;

}

void lecture() //Function 2

{

cout &lt;&lt; &quot;Lectures in the seminar on&quot; &lt;&lt; endl;

}

Seminar(int duration) //Function 3

{

time = duration;

cout &lt;&lt; &quot;Seminar starts now&quot; &lt;&lt; endl;

}

~Seminar() //Function 4

{

cout &lt;&lt; &quot;Thanks&quot; &lt;&lt; endl;

}

};

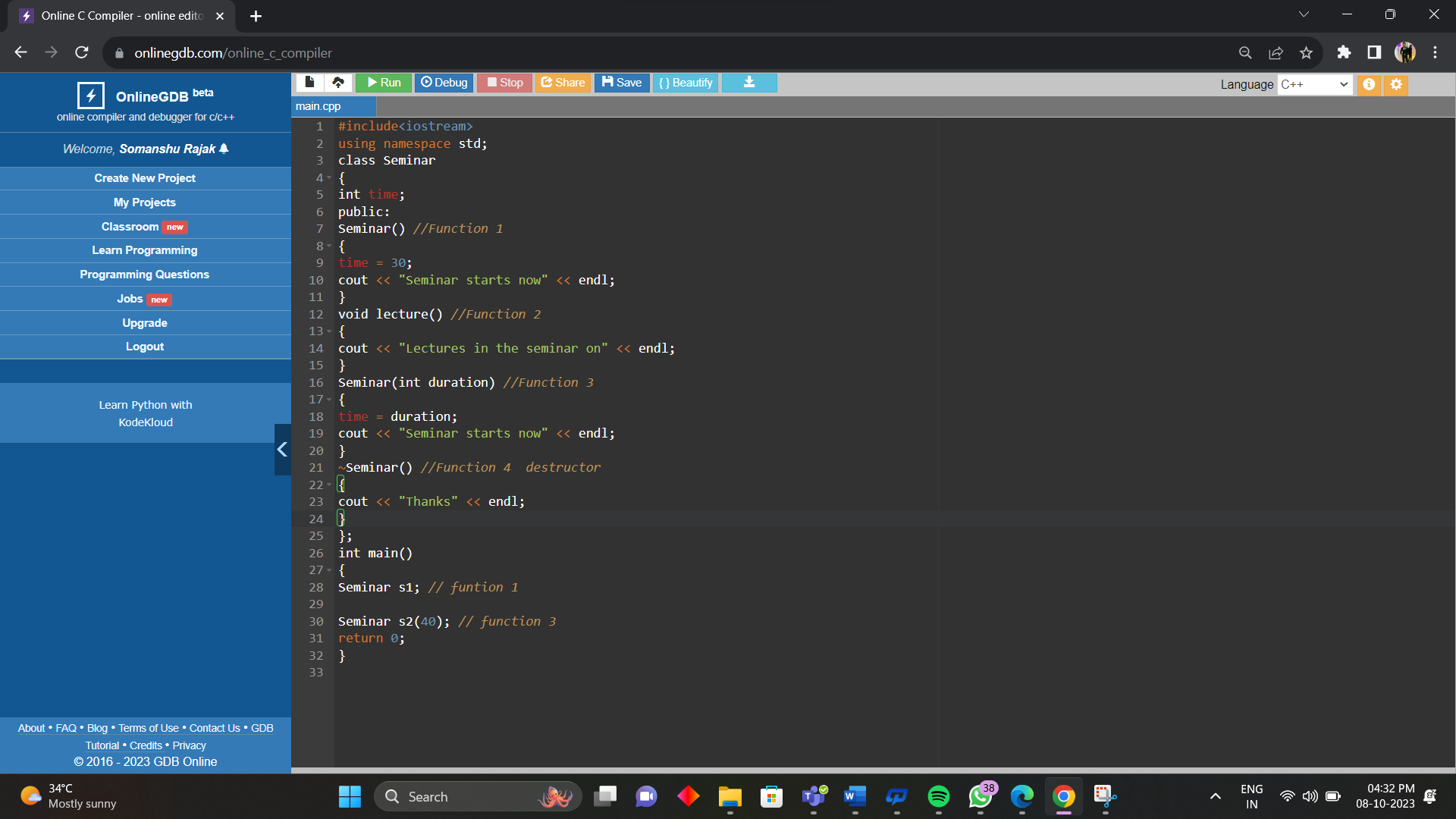
Statemants-:

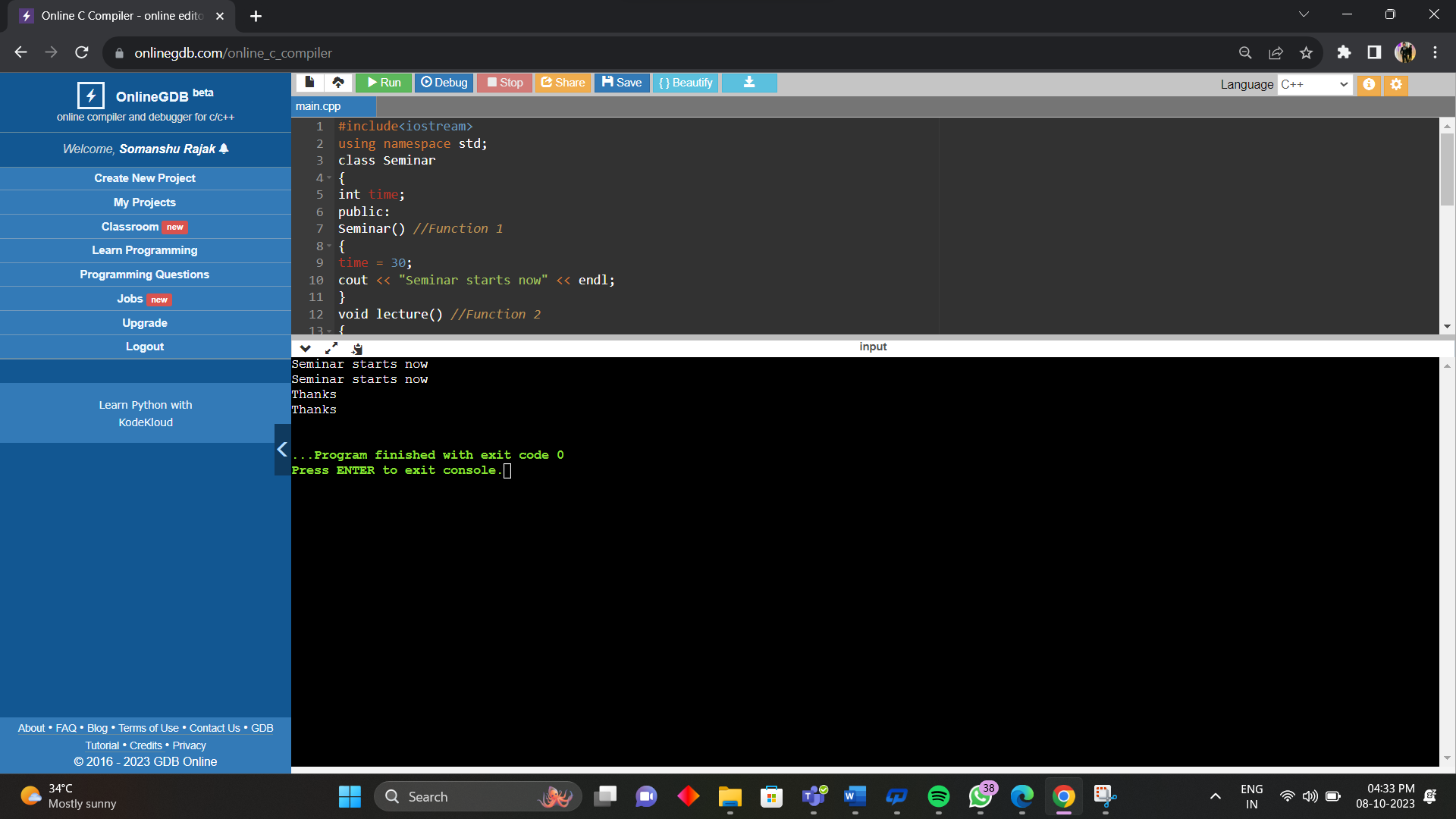
int main()

{

return 0;

}





i. Write statements in C++ that would execute Function 1 and Function 3 of class Seminar.

Ans-; Seminar s1;

s1.lecture();

Calls default constructor (Function 1)

Seminar s2(40);

s2.lecture();

Calls parameterized constructor (Function 3)

ii. In Object Oriented Programming, what is Function 4 referred as and when does it get invoked/called?

Ans-; Function 4 referred as Destructor

Function 4 will be automatically called when objects go out of scope (memory deallocate).

iii. In Object Oriented Programming, which concept is illustrated by Function 1 and Function 3 together?

Ans-; Function 1 as default constructor ,

Function 3 as parameterised construction one argument is passed in it

Constructor over-loading

2.Answer the questions (i) and (ii) after going through the following class:

class Test

{

char paper[20];

int marks;

public:

Test () // Function 1

{

strcpy (paper, &quot;Computer&quot;);

marks = 0;

}

Test (char p[]) // Function 2

{

strcpy(paper, p);

marks = 0;

}

Test (int m) // Function 3

{

strcpy(paper,&quot;Computer&quot;);

marks = m;

}

Test (char p[], int m) // Function 4

{

strcpy (paper, p);

marks = m;

}

};

int main()

{

Test F1; // Function 1

Test F2("Maths");// Function 2

Test F3(55); //Function 3

Test F4("Programming",50);// Function 4

return 0;

}

i. Write statements in C++ that would execute Function 1, Function 2, Function 3 and Function 4 of class Test.

Ans-;

Test F1; // Function 1

Test F2("Maths");// Function 2

Test F3(55); //Function 3

Test F4("Programming",50);// Function 4

ii. Which feature of Object Oriented Programming is demonstrated using Function

1, Function 2, Function 3 and Function 4 together in the above class Test?

Ans- Constructor Overloading

3.Consider the definition of the following class:

class Sample

{

private:

int x;

double y;

public :

Sample(); //Constructor 1

Sample(int); //Constructor 2

Sample(int, int); //Constructor 3

Sample(int, double); //Constructor 4

};

i. Write the definition of the constructor 1 so that the private member variables are initialized to 0.

Ans-

Sample :: Sample(); //initialized to value 0

{

x=0;

y=0;

}

ii. Write the definition of the constructor 2 so that the private member variable x is initialized according to the value of the parameter, and the private member variable y is initialized to 0.

Ans-

Sample :: Sample(int a)

{ x = a;

y = 0;

}

iii. Write the definition of the constructors 3 and 4 so that the private

member variables are initialized according to the values of the parameters.

Ans-

Sample :: Sample(int a,int b)

{ x = a;

y = b;

}

4. Create a class called Box with a variable: width of type double. Inside the class define a constructor and a friend that prints the width value(print Width). In the main() define a Box instance, set values and call printWidth.

#include <iostream>

using namespace std;

class Box

{

private:

double width;

public:

Box(double w)

{

width = w;

}

friend void printWidth(Box b);

};

void printWidth(Box b) {

cout << "The width of the box is " << b.width << endl;

}

int main()

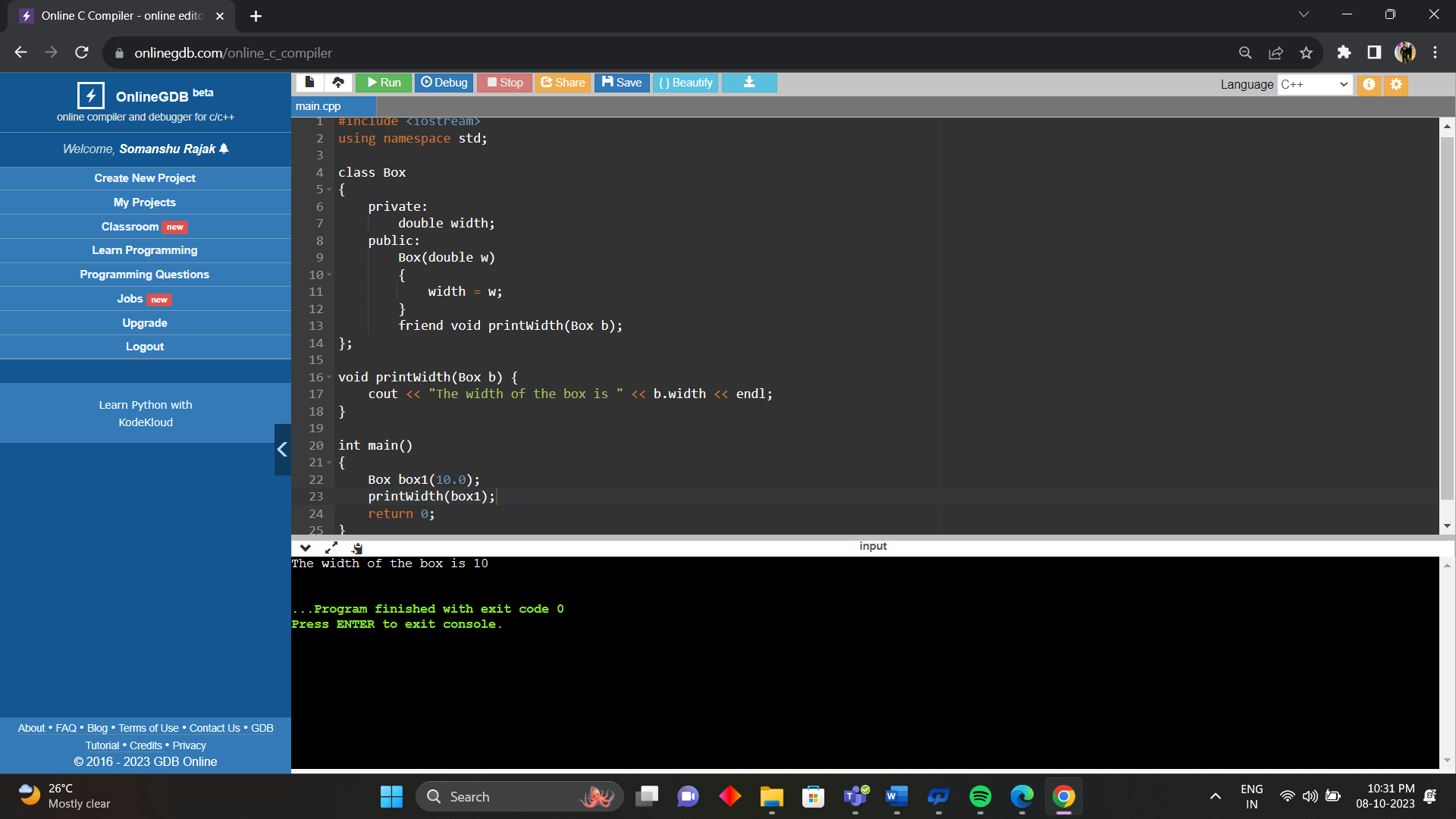
{

Box box1(10.0);

printWidth(box1);

return 0;

}



5. Perform addition operation on complex data using class and object. The program should ask for real and imaginary part of two complex numbers, and display the real and imaginary parts of their sum.

#include<iostream>

using namespace std;

class Complex{

public:

int real;

int imag;

void setvalue()

{

cout<<"Enter real complex number"<<endl;

cin>>real;

cout<<"Enter imag complex number"<<endl;

cin>>imag;

}

/\* Function to display the sum of two complex numbers \*/

void display()

{

cout<<real<<"+"<<imag<<"i"<<endl;

}

/\* Function to add two complex numbers \*/

void sum(Complex c1, Complex c2)

{

real=c1.real+c2.real;

imag=c1.imag+c2.imag;

}

};

int main()

{

Complex c1,c2,c3;

c1.setvalue();

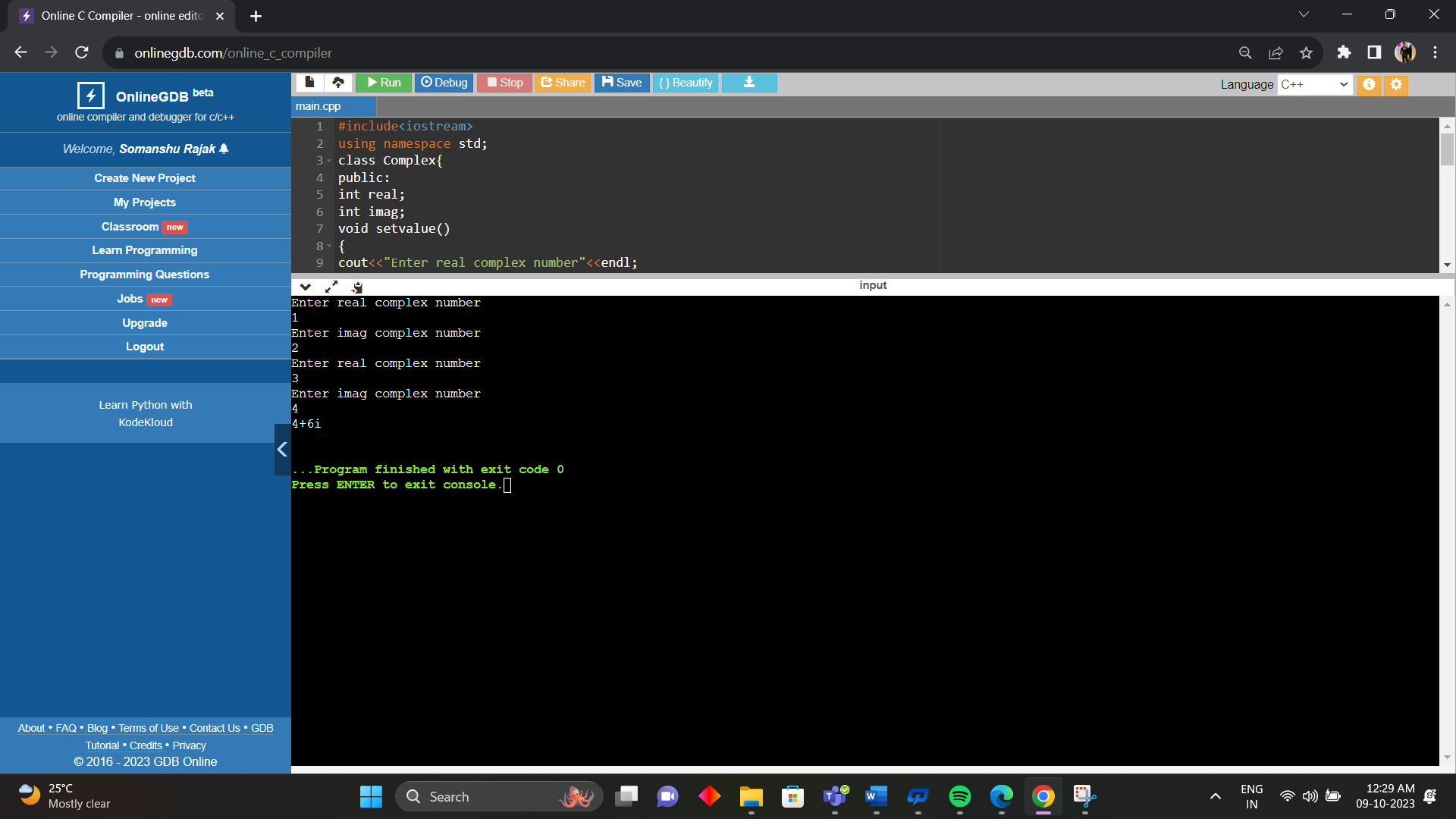
c2.setvalue();

c3.sum(c1,c2);

c3.display();

return 0;

}



6. Write a program in C++ having class string1 with members as Int length;Char \* buffer;Implement this class using copy constructor, destructor , parameterized constructor and default constructor.

#include <iostream>

#include <cstring>

using namespace std;

class string1 {

private:

int length;

char\* buffer;

public:

// Default constructor

string1() {

length = 0;

buffer = new char[length + 1];

buffer[0] = '\0';

}

// Parameterized constructor

string1(const char\* str) {

length = strlen(str);

buffer = new char[length + 1];

strcpy(buffer, str);

}

// Copy constructor

string1(const string1& str) {

length = str.length;

buffer = new char[length + 1];

strcpy(buffer, str.buffer);

}

// Destructor

~string1() {

delete[] buffer;

}

// Display the string

void display() {

cout << "String: " << buffer << endl;

cout << "Length: " << length << endl;

}

};

int main() {

// Default constructor

string1 s1;

// Parameterized constructor

string1 s2("Hello World!");

// Copy constructor

string1 s3(s2);

// Display the strings

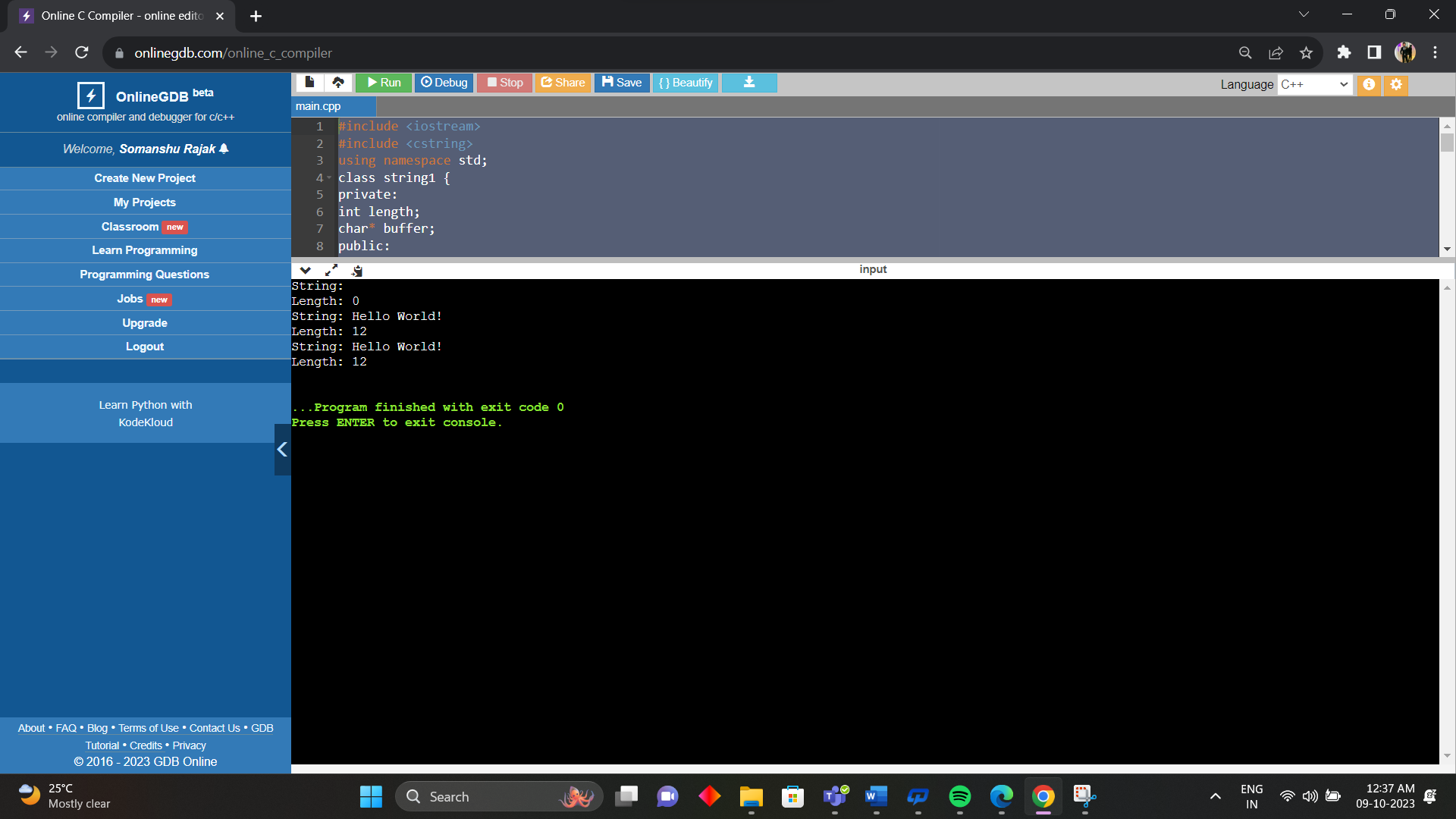
s1.display();

s2.display();

s3.display();

return 0;

}



7. Write a program in C++ that counts the number of objects created in the class. Illustrate the program with the constructor and destructor both .

#include <iostream>

using namespace std;

class Counting

{

private:

static int count;

public:

Counting()

{

count++;

cout << "Object created. Count = " << count << endl;

}

~Counting()

{

count--;

cout << "Object destroyed. Count = " << count << endl;

}

void display()

{

cout << "The current value of count is " << count << endl;

}

};

int Counting::count = 0;

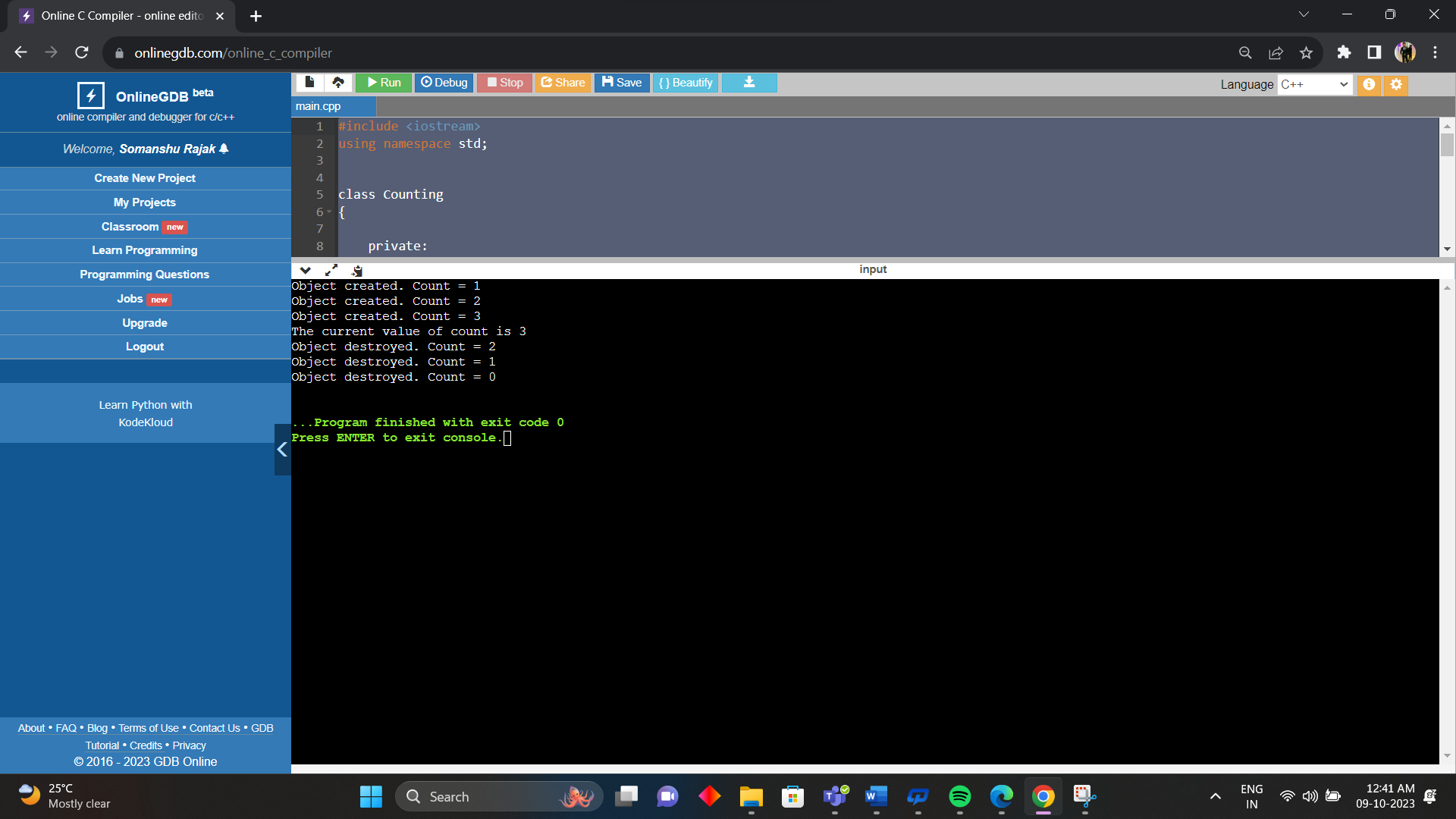
int main() {

Counting c1, c2, c3;

c1.display();

return 0;

}



8. Write a program in C++ to calculate perimeter of all figures using the concept of friend function.

#include <iostream>

using namespace std;

class Circle;

class Rectangle;

class Triangle;

class Shape {

public:

virtual double perimeter() = 0;

};

class Circle : public Shape {

private:

double radius;

public:

Circle(double r) {

radius = r;

}

double perimeter() {

return 2 \* 3.14 \* radius;

}

friend double totalPerimeter(Circle c, Rectangle r, Triangle t);

};

class Rectangle : public Shape {

private:

double length, breadth;

public:

Rectangle(double l, double b) {

length = l;

breadth = b;

}

double perimeter() {

return 2 \* (length + breadth);

}

friend double totalPerimeter(Circle c, Rectangle r, Triangle t);

};

class Triangle : public Shape {

private:

double a, b, c;

public:

Triangle(double x, double y, double z) {

a = x;

b = y;

c = z;

}

double perimeter() {

return a + b + c;

}

friend double totalPerimeter(Circle c, Rectangle r, Triangle t);

};

double totalPerimeter(Circle c, Rectangle r, Triangle t) {

return c.perimeter() + r.perimeter() + t.perimeter();

}

int main() {

Circle c(5);

Rectangle r(4, 6);

Triangle t(3, 4, 5);

cout << "Total perimeter: " << totalPerimeter(c, r, t) << endl;

return 0;

}

