**C++ CONSTRUCTOR ASSIGNMENT**

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;

}

};

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

Seminar.

**ANS: //** Seminar s1; (execute function 1)

Seminar s2(60); (execute function 3)//

ii. In Object Oriented Programming, what is Function 4 referred as and when does

it get invoked/called?

**ANS: //** Function 4 is called a destructor.

A destructor function is called automatically when the object goes out of scope:

(a) the function ends

(b) the program ends

(c) a block containing local variables ends

(d) a delete operator is called  //

iii. In Object Oriented Programming, which concept is illustrated by Function 1

and Function 3 together?

**ANS: //** Constructor Overloading (Polymorphism).//

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;

}

};

i. Write statements in C++ that would execute Function 1, Function 2, Function 3

and Function 4 of class Test.

**ANS: //**

Test t1; // Execute funtion 1

Test t2("Hindi"); // Execute funtion 2

Test t3(40); // Execute funtion 3

Test t4("Science",52); // Execute funtion 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 (Polymorphism)

**//**

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()

{

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;

}

Sample :: Sample(int a, double b)

{

x = a;

y = b;

}**//**

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 {

private:

    double real;

    double imagi;

public:

    Complex() : real(0.0), imagi(0.0) {}

    Complex(double r, double i) : real(r), imagi(i) {}

    Complex add(const Complex& other) {

        Complex result;

        result.real = real + other.real;

        result.imagi = imagi + other.imagi;

        return result;

    }

    void display() {

        cout << "Sum: " << real << " + " << imagi << "i" << endl;

    }

};

int main() {

    double real1, imagi1, real2, imagi2;

    cout << "Enter the real and imaginary parts of first complex number:" << endl;

    cin >> real1 >> imagi1;

    cout << "Enter the real and imaginary parts of  second complex number:" << endl;

    cin >> real2 >> imagi2;

    Complex complex1(real1, imagi1);

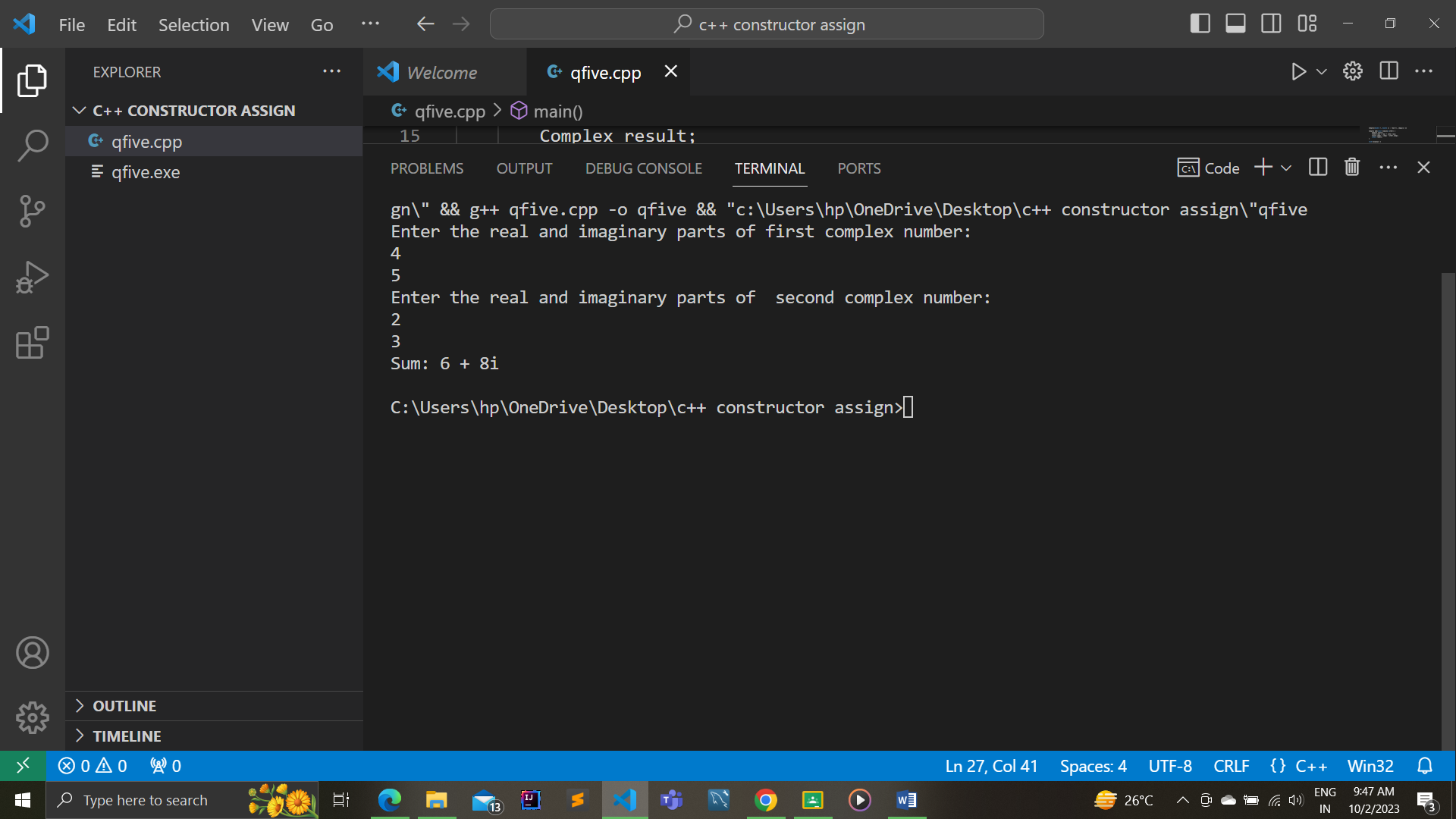
    Complex complex2(real2, imagi2);

    Complex sum = complex1.add(complex2);

    sum.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 <string.h>

using namespace std;

class String1 {

private:

    int length;

    char\* buffer;

public:

    // Default constructor

    String1() {

        length = 0;

        buffer = nullptr;

    }

    // Parameterized constructor

    String1(const char\* str) {

        length = strlen(str);

        buffer = new char[length + 1];

        strcpy(buffer, str);

    }

    // Copy constructor

    String1(const String1& other) {

        length = other.length;

        buffer = new char[length + 1];

        strcpy(buffer, other.buffer);

    }

    // Destructor

    ~String1() {

        delete[] buffer;

    }

    // Member function to display the string

    void display() {

        if (buffer != nullptr) {

            cout << buffer << endl;

        } else {

            cout << "Empty String" << endl;

        }

    }

};

int main() {

    String1 str1;

    cout << "String1: ";

    str1.display();

    String1 str2("Hello,Bhopal!");

    cout << "String2: ";

    str2.display();

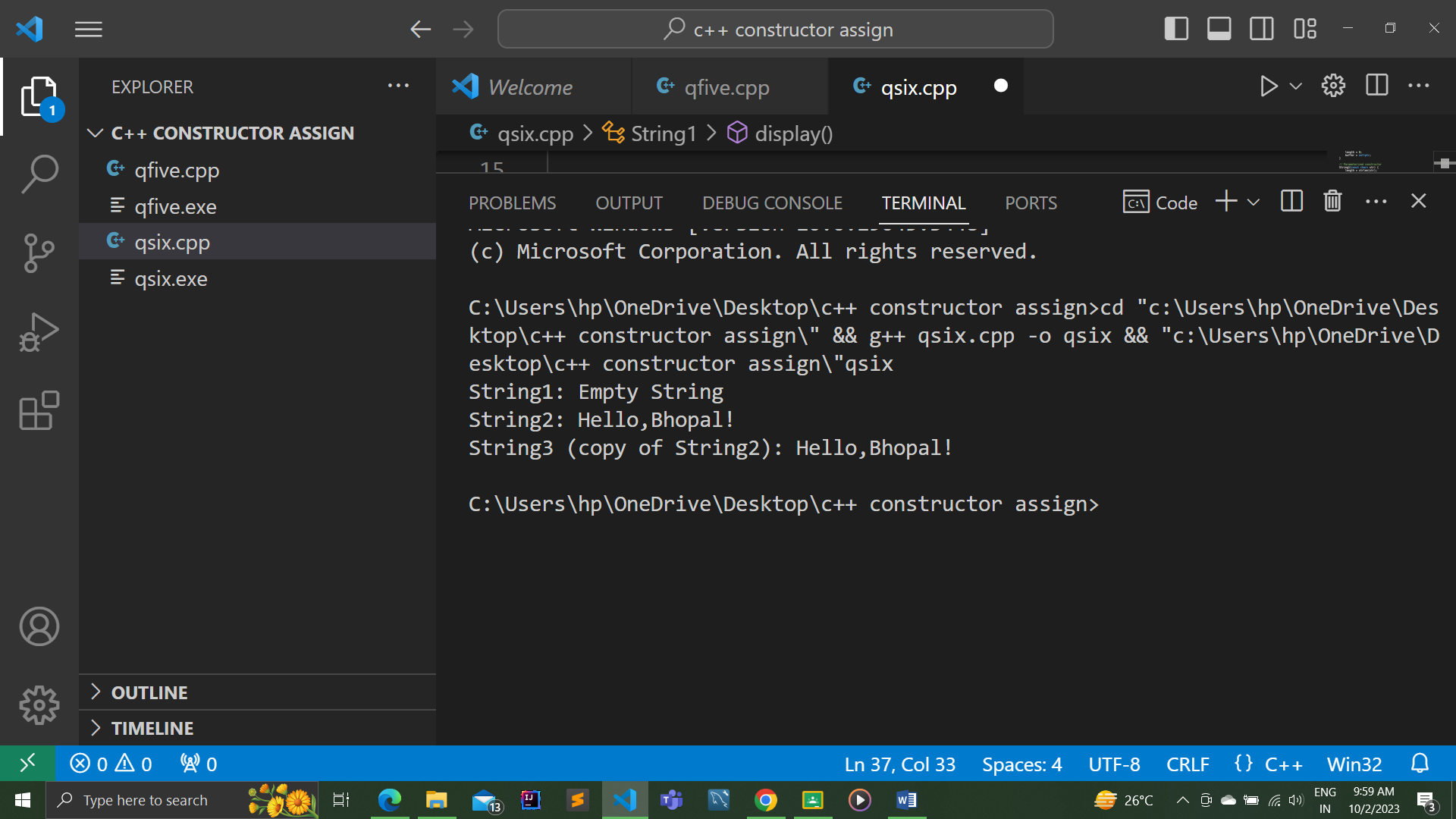
    String1 str3 = str2;

    cout << "String3 (copy of String2): ";

    str3.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 Laptop {

private:

    static int count;

public:

    Laptop() {

        count++;

        cout << "Laptop object created. Total count: " << count << endl;

    }

    ~Laptop() {

        count--;

        cout << "Laptop object destroyed. Total count: " << count << endl;

    }

    static int getCount() {

        return count;

    }

};

int Laptop::count = 0;

int main() {

    Laptop laptop1;

    Laptop laptop2;

    Laptop laptop3;

    cout << "Total laptops created: " << Laptop::getCount() << endl;

    // Creating more objects

    Laptop laptop4;

    Laptop laptop5;

    cout << "Total laptops created: " << Laptop::getCount() << endl;

    return 0;

}

