Lab Manual:-

Lab Task No 1 :-

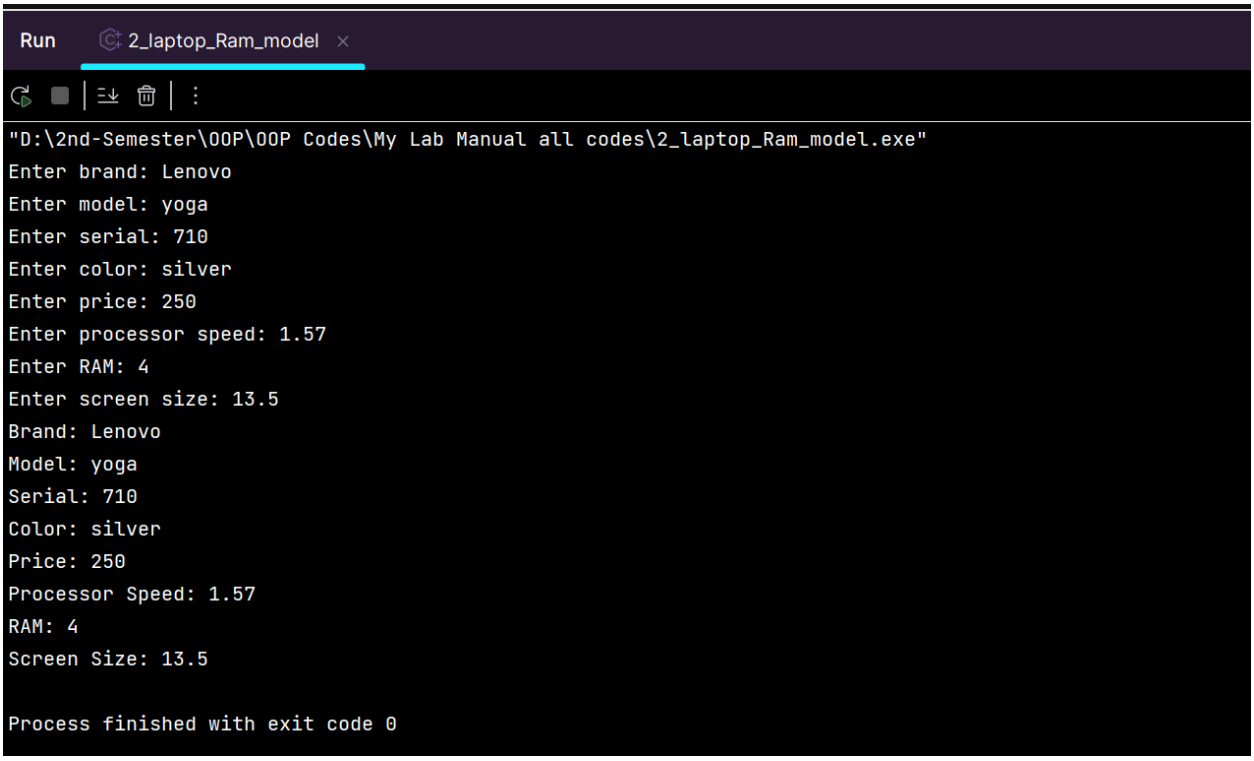
Write a C++ program that creates a class called laptop. The data members of the class are brand (string), model (string), serial (int), color (string), price (float), processor speed (float), RAM (int), screen size(float). Create member function that will set the individual values. Since the RAM can be upgraded therefore create a function that allows you to upgrade the RAM only. In the end, create a function that will display all the data members.

```
//
// Created by zohaib on 29/05/2023.
//
#include<iostream>
using namespace std;


class Laptop{
private:
    string brand;
    string model;
    string serial;
    string color;
    float price;
    float processor_speed;
    int RAM;
    float screen_size;
public:
    void set_values(){
        cout<<"Enter brand: ";
        cin>>brand;
```

```
        cout<<"Enter model: ";
        cin>>model;
        cout<<"Enter serial: ";
        cin>>serial;
        cout<<"Enter color: ";
        cin>>color;
        cout<<"Enter price: ";
        cin>>price;
        cout<<"Enter processor speed: ";
        cin>>processor_speed;
        cout<<"Enter RAM: ";
        cin>>RAM;
        cout<<"Enter screen size: ";
        cin>>screen_size;
    }
    void display_values(){
        cout<<"Brand: "<<brand<<endl;
        cout<<"Model: "<<model<<endl;
        cout<<"Serial: "<<serial<<endl;
        cout<<"Color: "<<color<<endl;
        cout<<"Price: "<<price<<endl;
        cout<<"Processor Speed: "<<processor_speed<<endl;
        cout<<"RAM: "<<RAM<<endl;
        cout<<"Screen Size: "<<screen_size<<endl;
    }
};
int main(){
    Laptop xyz;
    xyz.set_values();
    xyz.display_values();
    return 0;
```

```
}
```

Output:-



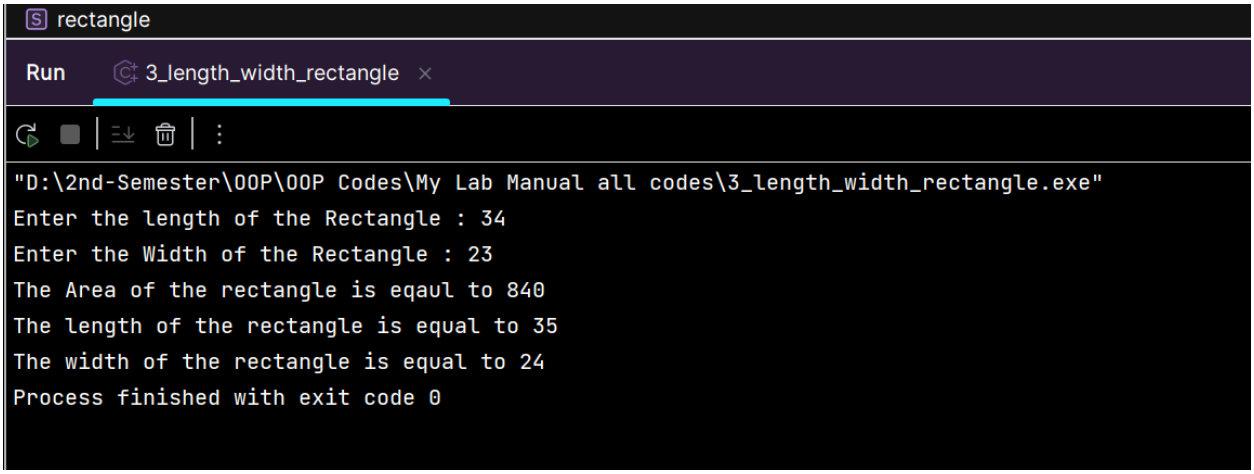
Lab Task No 2 :-

Write a class called rectangle. Your task is to store the length and width of the rectangle. Write a member function called increment that will add 1 to the value of length and width. Also write a function that will compute the area of the rectangle. Finally write a constant function that will display the length, width and area of the rectangle. Demonstrate the use of the object in the main function. Make sure that the function names are meaningful.

```
//
// Created by zohaib on 29/05/2023.
//
#include <iostream>
using namespace std;

class rectangle{
private:
    float length, width;
public:
    rectangle(){
        cout<<"Enter the length of the Rectangle : ";
        cin>>length;
```

```
        cout<<"Enter the Width of the Rectangle : ";
        cin>>width;
        length++;
        width++;
    }
    float calArea(){

        return length*width;
    }
    void display(){
        cout<<"The Area of the rectangle is eqaul to "<< calArea();
        cout<<"\nThe length of the rectangle is equal to "<<length;
        cout<<"\nThe width of the rectangle is equal to "<<width;
    }
};
int main(){
    rectangle qwe;
    qwe.display();
    return 0;
}
```

Output:-



Lab Task No 3 :-

Write a program that creates a class called number. Your class will have two data members namely num (float) and result (int). To find the factorial of the entered number you will need to design three functions as follows:

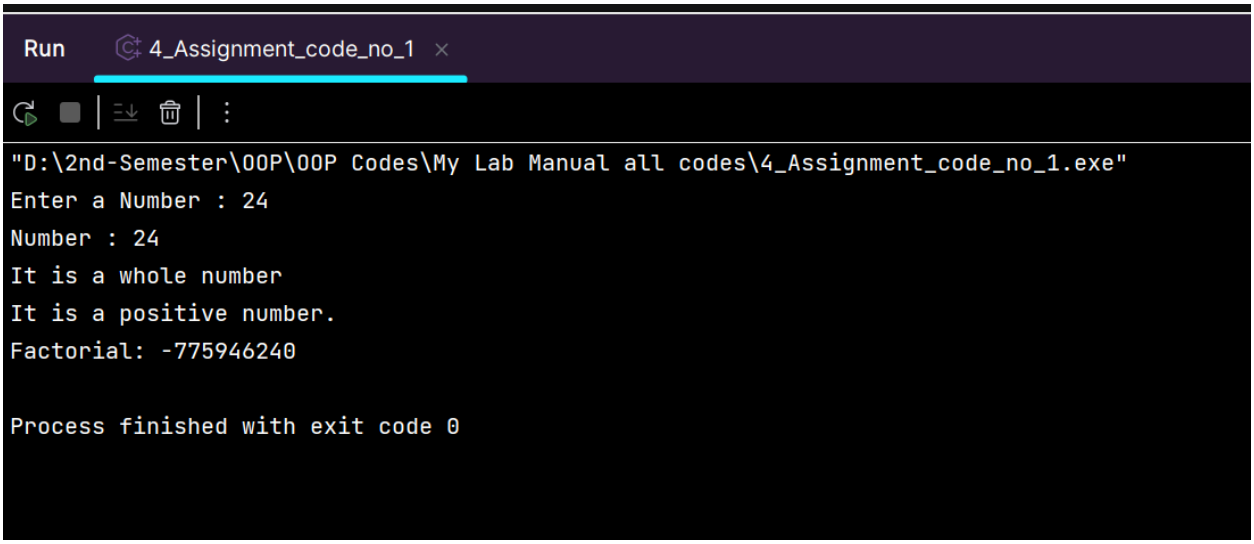
1. Function to determine if a number is a whole number or not
2. Function to determine if the number is positive or not
3. Function to find the actual factorial
4. Function to display the number and its factorial Remember that to find the factorial the number must of positive and a whole number.

5. So if any of these conditions are not met then you cannot determine the factorial.

```
//
// Created by zohaib on 29/05/2023.
//

#include<iostream>
using namespace std;

class Number {
private:
    float num; // private data member to store the input number
    //  int result;

public:
    // Function to check if number is a whole number
    bool WholeNum() {
        int wholenum = (int)num; // Typecast num to an int to remove decimal
part
        if (num == wholenum){//checks if the original number is equal to the
original value of the number
            return true; // returns true if number is a whole number
        }
        else{
            return false; // returns false if number is not a whole number
        }
    }

    // Function to check if number is positive
    bool PositiveNum() {
        if(num > 0){
            return true; // returns true if number is positive
        }
        else {
            return false; // returns false if number is not positive
        }
    }

    // Function to calculate the factorial of the given number
    int FactCal() {
        int fact = 1;
        for(int i = 1; i <= num; i++) {
            fact *= i; // calculates the factorial of the number
        }
        return fact; // returns the calculated factorial
    }

    // Function to display the result
    void Displayresult() {
        cout << "Number : " << num << endl; // displays the input number
```

```
        if(WholeNum()){
            cout << "It is a whole number " << endl; // displays if number is
a whole number
        }
        else {
            cout << "It is not a whole number." << endl; // displays if
number is not a whole number
        }
        if(PositiveNum()) {
            cout << "It is a positive number." << endl; // displays if number
is positive
        }
        else {
            cout << "It is not a positive number." << endl; // displays if
number is not positive
        }
        cout << "Factorial: " << FactCal() << endl; // displays the
calculated factorial
    }

    // Function to set the value of num
    void setNum(float n) {
        num = n; // sets the value of input number
    }
};

int main(){
    float num;
    cout << "Enter a Number : ";
    cin >> num;

    Number xyz; // creates an object of the Number class
    xyz.setNum(num); // sets the input number in the object
    xyz.Displayresult(); // calls the Displayresult() function to display the
result
    return 0;
}
```

Output:-



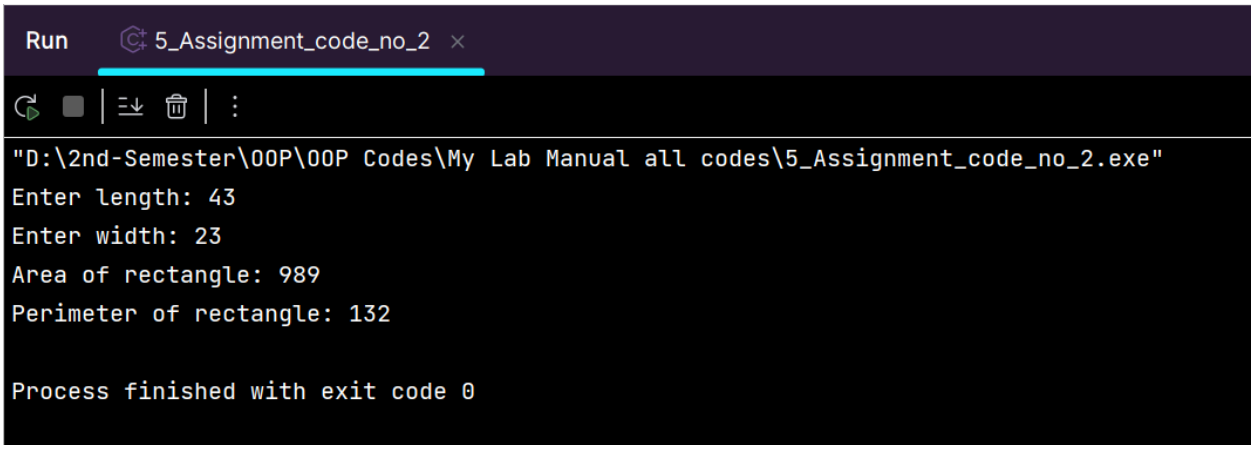
Lab Task No 4 :-

Design a program that uses the "Geometry" class. The program should prompt the user to enter the length and width of a shape. If both values are equal, the program should call the "square" function to calculate the area and perimeter of a square. Otherwise, it should call the "rectangle" function to calculate the area and perimeter of a rectangle.

```
//
// Created by zohaib on 29/05/2023.
//

#include <iostream>
using namespace std;

// Define a class called Geometry
class Geometry {
private:
    float length, width; // Declare two private data members representing the
length and width of the shape

public:
    // Defining a constructor to initialize the data members of the object
    Geometry() {
        cout << "Enter length: ";
        cin >> length;
        cout << "Enter width: ";
        cin >> width;
    }

    // A function to calculate and display the area and perimeter of a square
    void square() {
```

```
        float area = length * length;
        float perimeter = 4 * length;
        cout << "Area of square: " << area << endl;
        cout << "Perimeter of square: " << perimeter << endl;
    }

    // A function to calculate and display the area and perimeter of a
rectangle
    void rectangle() {
        float area = length * width;
        float perimeter = 2 * (length + width);
        cout << "Area of rectangle: " << area << endl;
        cout << "Perimeter of rectangle: " << perimeter << endl;
    }

    // A function to display the area and perimeter based on whether it is a
square or a rectangle
    void display(){
        if (length == width) {
            square();
        } else {
            rectangle();
        }
    }
};

// Main function
int main(){

    // Create an object of the Geometry class with the given length and width
    Geometry xyz;

    // Display the area and perimeter of the shape
    xyz.display();

    // End the program and return 0
    return 0;
}
```

Output:-



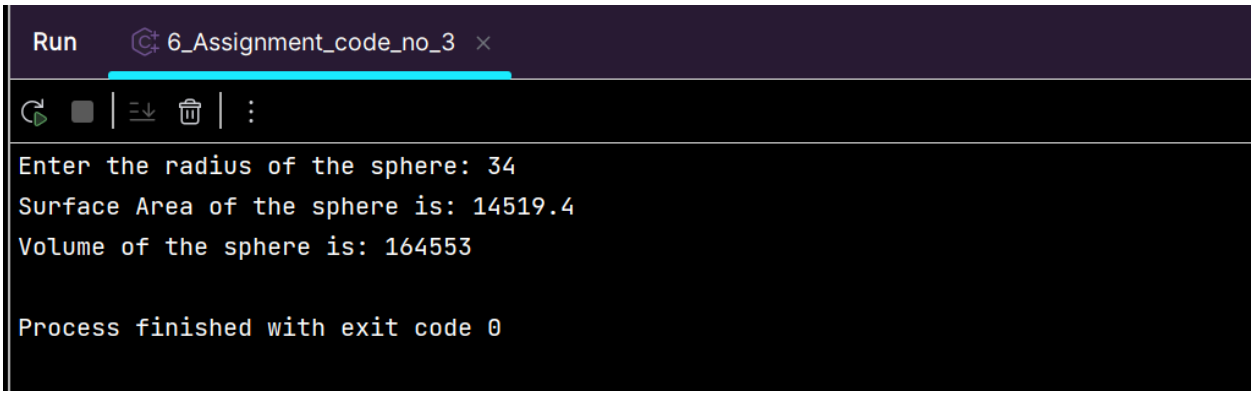
Lab Task No 5 :-

Write a C++ Program to implement a sphere class with appropriate members and member function to find the surface area and the volume. (Surface = 4 π r2 and Volume = 4/ 3 π r3).

```
//
// Created by zohaib on 29/05/2023.
//
/*

#include <iostream>
using namespace std;

class Sphere {
private:
    float radius;

public:
    // Constructor to take input from the user and initialize the value of
radius
    Sphere(){
        cout << "Enter the radius of the sphere: ";
        cin >> radius;
    }

    // Function to calculate the surface area of the sphere and return the
value
    float SurfaceArea() {
        float surfaceArea = 4 * 3.14 * radius * radius;
        return surfaceArea;
    }

    // Function to calculate the volume of the sphere and return the value
    float Volume() {
        float volume = (4.0 / 3.0) * 3.14 * radius * radius * radius;
        return volume;
```

```
    }
};

int main() {
    // Create an object of the Sphere class
    Sphere xyz;

    // Call the SurfaceArea function and display the output
    cout << "Surface Area of the sphere is: " << xyz.SurfaceArea() << endl;

    // Call the Volume function and display the output
    cout << "Volume of the sphere is: " << xyz.Volume() << endl;

    return 0;
}
```

Output:-



Lab Task No 6 :-

Design a menu-driven program that enables users to perform arithmetic operations on two numbers. The program should provide the options to add (+), subtract (-), multiply (*), or divide (/) and prompt users to input the numbers. In addition, the program should have the following functions:

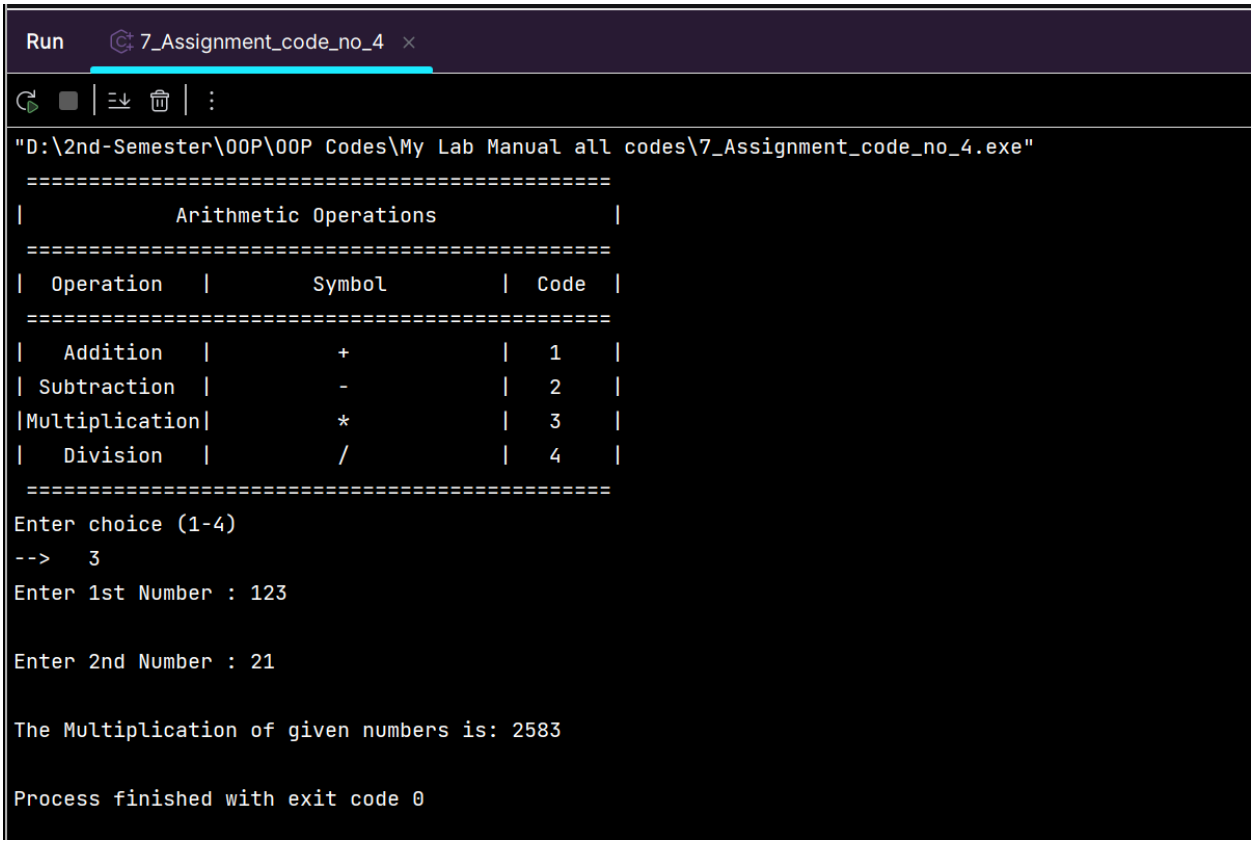
1. "show-Choice" function that displays the options available to the user and provides instructions on how to enter the data.
2. "add" function that takes two arguments as input and returns their sum.
3. "subtract" function that takes two arguments as input and returns their difference.
4. "multiply" function that takes two arguments as input and returns their product.
5. "divide" function that takes two arguments as input and returns their quotient.

```
//
// Created by zohaib on 29/05/2023.
//
```

```
#include<iostream>
using namespace std;

class Calculator {
private:
    float num1, num2;
public:

    Calculator() {
        int choice;
        cout << " ================================================" << endl;
        cout << "|            Arithmetic Operations              |" << endl;
        cout << " ================================================" << endl;
        cout << "|  Operation   |        Symbol         |  Code  |" << endl;
        cout << " ================================================" << endl;
        cout << "|   Addition   |          +            |   1    |" << endl;
        cout << "| Subtraction  |          -            |   2    |" << endl;
        cout << "|Multiplication|          *            |   3    |" << endl;
        cout << "|   Division   |          /            |   4    |" << endl;
        cout << " ================================================" << endl;
        cout<<"Enter choice (1-4)\n";
        cout<<"-->   ";
        cin >> choice;
        cout << "Enter 1st Number : ";
        cin >> num1;
        cout<<endl;
        cout << "Enter 2nd Number : ";
        cin >> num2;
        cout<<endl;

        switch(choice) {
            case 1: {
                cout << "The addition of given numbers is: " << add(num1,
num2) << endl;
                break;
            }
            case 2: {
                cout << "The Substraction of given numbers is: " << sub(num1,
num2) << endl;
                break;
            }
            case 3: {
                cout << "The Multiplication of given numbers is: " <<
multiply(num1, num2) << endl;
                break;
            }
            case 4: {
                cout << "The Division of the given numbers is: " << div(num1,
```

```
num2) << endl;
                break;
            }
            default:{
                cout<<"Invalid choice";
                break;
            }
        }
    }

    float add(float num1, float num2) {
        return (num1 + num2);
    }
    float sub(float num1, float num2) {
        return (num1 - num2);
    }
    float multiply(float num1, float num2) {
        return (num1 * num2);
    }
    float div(float num1, float num2) {
        return (num1 / num2);
    }
};

int main() {
    Calculator xyz;
    return 0;
}
```

Output:-



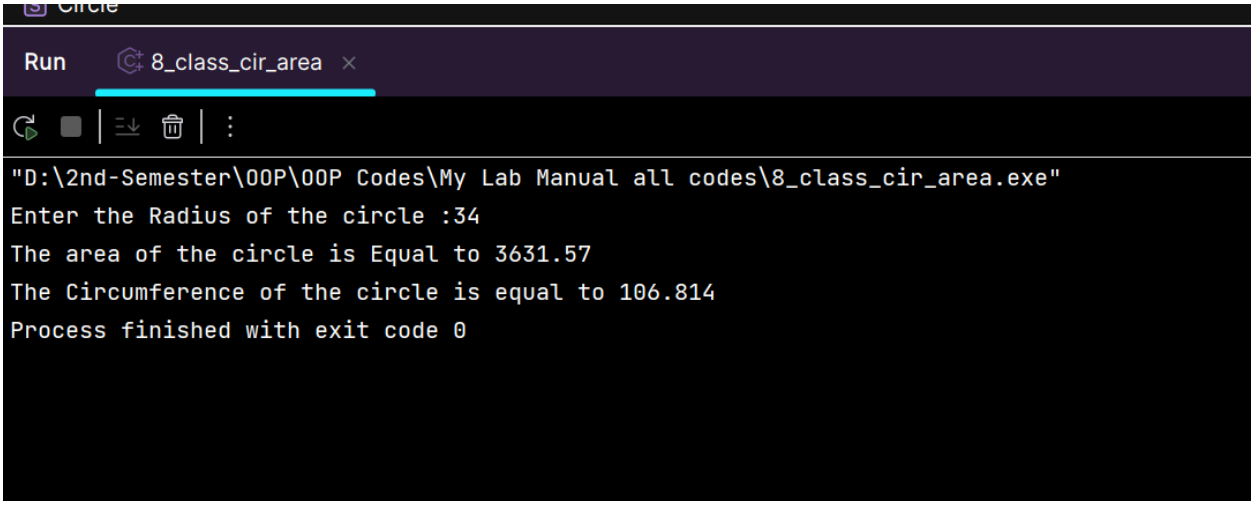
Lab Task No 7 :-

Create a Class named as Circle. Create a data member named as radius. Create member functions to calculate area of circle and circumference of the circle.

```
//
// Created by zohaib on 29/05/2023.
//

#include <iostream>
using namespace std;

class Circle{
private:
    float radius;
public:
    void getter(){
        cout<<"Enter the Radius of the circle :";
        cin>>radius;
    }
    void calArea(){
        cout<<"The area of the circle is Equal to "<<3.1415*radius*radius;
    }
```

```
    void calCir(){
        cout<<"\nThe Circumference of the circle is equal to
"<<3.1416*radius;
    }
};

int main(){
    Circle xyz;
    xyz.getter();
    xyz.calArea();
    xyz.calCir();
    return 0;
}
```

Output:-



Lab Task No 8 :-

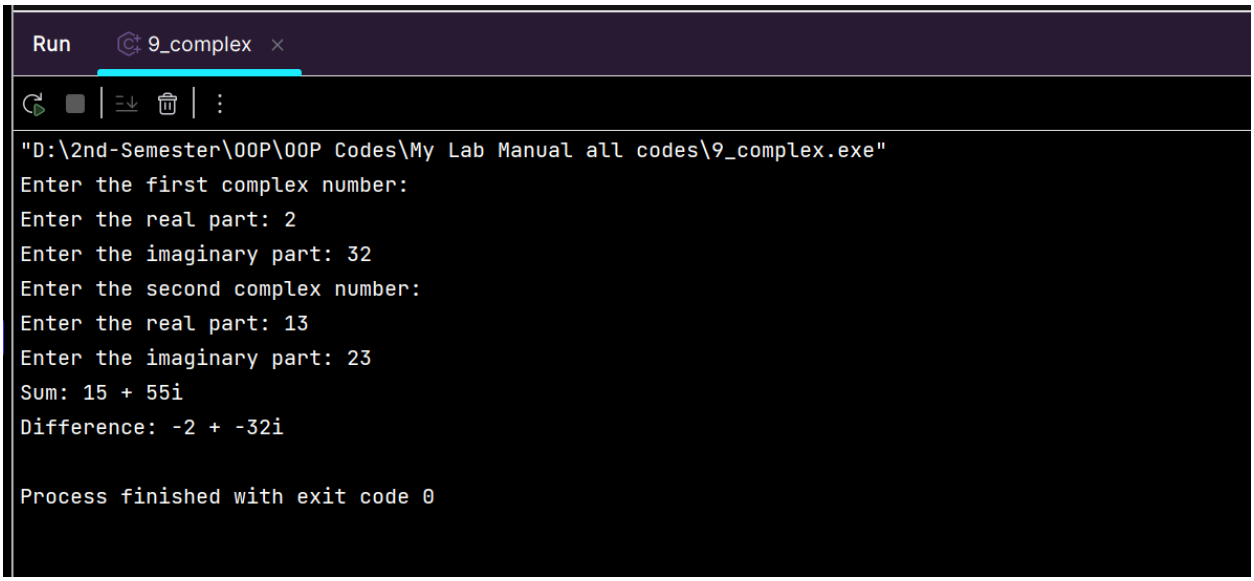
Create a Class named as Complex. Create Data members named as real and imaginary.

1. Create a function to get values from the user.
2. Create a function to add two complex numbers.
3. Create a function to subtract two complex numbers.

```
//
// Created by zohaib on 29/05/2023.
//

#include <iostream>

using namespace std;

class Complex {
private:
```

```
    double real;
    double imaginary;

public:
    void getValues() {
        cout << "Enter the real part: ";
        cin >> real;
        cout << "Enter the imaginary part: ";
        cin >> imaginary;
    }

    void add(Complex other) {
        real += other.real;
        imaginary += other.imaginary;
    }

    void subtract(Complex other) {
        real -= other.real;
        imaginary -= other.imaginary;
    }

    double getReal() const {
        return real;
    }

    double getImaginary() const {
        return imaginary;
    }
};

int main() {
    Complex xyz1, xyz2;
    cout << "Enter the first complex number:\n";
    xyz1.getValues();
    cout << "Enter the second complex number:\n";
    xyz2.getValues();

    xyz1.add(xyz2);
    xyz2.subtract(xyz1);

    cout << "Sum: " << xyz1.getReal() << " + " << xyz1.getImaginary() <<
"i\n";
    cout << "Difference: " << xyz2.getReal() << " + " << xyz2.getImaginary()
<< "i\n";

    return 0;
}
```

Output:-



Lab Task No 9 :-

Write the definition of a class person. Add at least 5 attributes (private/public) to the person class. Add the following behavior (function) to the class:

getName: returns name

setName: sets the name to the passed string

getAge: returns age

setAge: sets the age to the passed integer

isMale: returns true or false

isFemale: returns true or false

getOccupation: returns a string

canCook : returns true or false

Create different objects of class person which uses different version of constructors and access all the methods of the class and show the result of each method call in proper form.

Default initialize all the attributes of the class. Also define parameterize and copy constructor of class

Default values:

Name: “”

Age=0

Gender=m (m for male, f for female)

Occupation= student

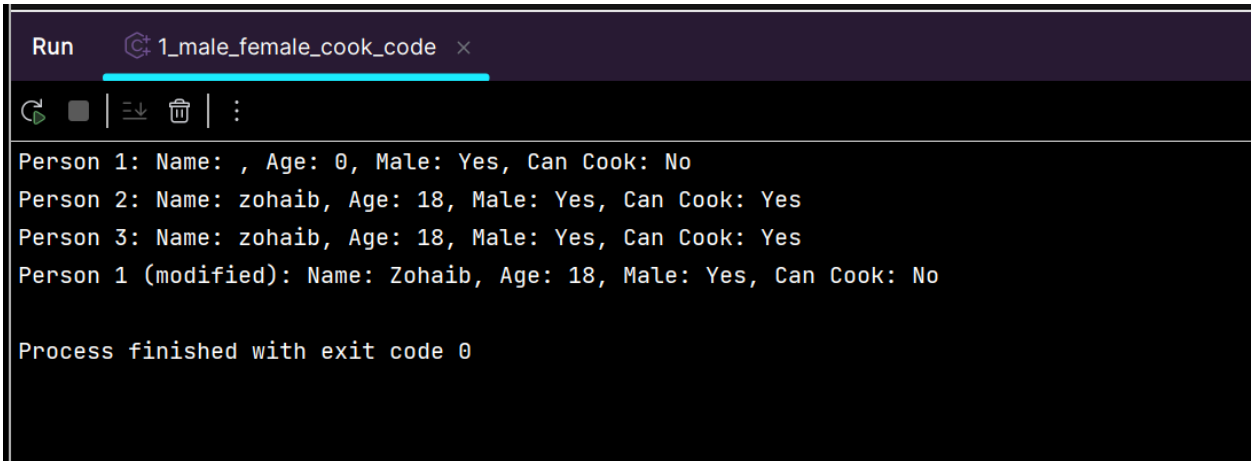
Cooking= n (n for no, y for yes)

```
//
// Created by zohaib on 29/05/2023.
//

#include <iostream>
#include <string>
using namespace std;

class Person {
private:
    string name;
    int age;
    char gender;
    string occupation;
    char canCook1;

public:
    // Default constructor
    Person() {
        name = "";
        age = 0;
        gender = 'm';
        occupation = "student";
        canCook1 = 'n';
    }

    // Parameterized constructor
    Person(string n, int a, char g, string o, char c) {
        name = n;
        age = a;
        gender = g;
        occupation = o;
        canCook1 = c;
    }

    // Copy constructor
    Person(const Person& other) {
        name = other.name;
        age = other.age;
        gender = other.gender;
```

```
        occupation = other.occupation;
        canCook1 = other.canCook1;
    }

    string getName() {
        return name;
    }

    void setName(string n) {
        name = n;
    }

    int getAge() {
        return age;
    }

    void setAge(int a) {
        age = a;
    }

    bool isMale() {
        return gender == 'm';
    }

    bool isFemale() {
        return gender == 'f';
    }

    string getOccupation() {
        return occupation;
    }

    bool canCook() {
        return (canCook1 == 'y' || canCook1 == 'Y');
    }
};

int main() {
    // Create objects using different constructors
    Person person1;  // Default constructor
    Person person2("zohaib", 18, 'm', "student", 'y');  // Parameterized
constructor
    Person person3 = person2;  // Copy constructor

    // Access methods and display results
    cout << "Person 1: Name: " << person1.getName() << ", Age: " <<
person1.getAge() << ", Male: " << (person1.isMale() ? "Yes" : "No") << ", Can
Cook: " << (person1.canCook() ? "Yes" : "No") << endl;
    cout << "Person 2: Name: " << person2.getName() << ", Age: " <<
```

```
person2.getAge() << ", Male: " << (person2.isMale() ? "Yes" : "No") << ", Can
Cook: " << (person2.canCook() ? "Yes" : "No") << endl;
    cout << "Person 3: Name: " << person3.getName() << ", Age: " <<
person3.getAge() << ", Male: " << (person3.isMale() ? "Yes" : "No") << ", Can
Cook: " << (person3.canCook() ? "Yes" : "No") << endl;

    // Modify attributes using setter methods
    person1.setName("Zohaib");
    person1.setAge(18);

    // Display modified attributes
    cout << "Person 1 (modified): Name: " << person1.getName() << ", Age: "
<< person1.getAge() << ", Male: " << (person1.isMale() ? "Yes" : "No") << ",
Can Cook: " << (person1.canCook() ? "Yes" : "No") << endl;

    return 0;
}
```

Output:-



Lab Task No 10 :-

Create a class called time that has separate private int member data for hours, minutes, and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. Another member function should display it, in 11:59:59 format. The final member function should add two objects of type time passed as arguments. A main() program should create two initialized time objects and one that isn't initialized. Then it should add the two initialized values together, leaving the result in the third time

variable. Finally it should display the value of this third variable. Also write output of your program and values you supposed to add.

```
#include <iostream>

using namespace std;

class Time {
private:
    int hours;
    int minutes;
    int seconds;

public:
    // Default constructor
    Time() {
        hours = 0;
        minutes = 0;
        seconds = 0;
    }

    // Constructor to initialize with fixed values
    Time(int h, int m, int s) {
        hours = h;
        minutes = m;
        seconds = s;
    }

    void displayTime() {
        cout << hours << ":" << minutes << ":" << seconds << endl;
    }

    void addTime(const Time& other) {
        seconds += other.seconds;
        minutes += seconds / 60;
        seconds %= 60;

        minutes += other.minutes;
        hours += minutes / 60;
        minutes %= 60;

        hours += other.hours;
    }
};
```

```
int main() {
    Time t1(2, 30, 45); // 02:30:45
    Time t2(1, 15, 30); // 01:15:30
    Time t3;            // Default-initialized to 00:00:00

    cout << "Time 1: ";
    t1.displayTime();

    cout << "Time 2: ";
    t2.displayTime();

    t3.addTime(t1);
    t3.addTime(t2);

    cout << "Resultant Time: ";
    t3.displayTime();

    return 0;
}
```

Output:-

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS C:\Users\zohai> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
Time 1: 2:30:45
Time 2: 1:15:30
Resultant Time: 3:46:15
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Lab Task No 11 :-

Write a program which takes values in an array and stores it in a pointer, then that pointer prints values of the array.

```
#include <iostream>
using namespace std;

class Array {
private:
    int* arr;
    int size;
```

```
public:
    Array(int size) {
        this->size = size;
        arr = new int[size];
    }


    void inputValues() {
        for (int i = 0; i < size; i++) {
            cout << "Enter value for element " << i << ": ";
            cin >> arr[i];
        }
    }

    void printValues() const {
        cout << "Array values: ";
        for (int i = 0; i < size; i++) {
            cout << arr[i] << endl;
        }
        cout << endl;
    }
};

int main() {
    int size;
    cout << "Enter the size of the array: ";
    cin >> size;

    Array obj(size);

    obj.inputValues();
    obj.printValues();

    return 0;
}
```

Output:-

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual a
 }
Enter the size of the array: 5
Enter value for element 0: 1
Enter value for element 1: 2
Enter value for element 2: 3
Enter value for element 3: 4
Enter value for element 4: 5
Array values: 1
2
3
4
5
```

Lab Task No 12 :-

Write a program which takes a string as an input and stores that string to a pointer. Then the user inputs if the character is there or not. If it's there, then it prints the character, else it says no character found.

```
#include <iostream>
#include <cstring>

using namespace std;

class StringSearch {
private:
    char str[100]; // Assuming a maximum length of 100 characters
    char* ptr;

public:
    StringSearch() {
        cout << "Enter a string: ";
        cin.getline(str, sizeof(str));
        ptr = str;
    }

    void searchCharacter() {
        char target;
        cout << "Enter a character to search for: ";
        cin >> target;

        bool found = false;
        while (*ptr != '\0') {
            if (*ptr == target) {
                found = true;
                break;
```

```
            }
            ptr++;
        }

        if (found) {
            cout << "Character '" << target << "' found." << endl;
        } else {
            cout << "Character '" << target << "' not found." << endl;
        }
    }
};

int main() {
    StringSearch search;

    search.searchCharacter();

    return 0;
}
```

Output:-

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
Enter a string: Zohaib
Enter a character to search for: a
Character 'a' found.
```

Lab Task No 13 :-

Create a file using fstream.

```
#include <iostream>
#include <fstream>
using namespace std;

class File {
private:
    ofstream MyFile;
public:
```

```
    void MakeFile() {
        MyFile.open("file.txt");
        if (MyFile.is_open()) {
            MyFile << " Zohaib Khalid ";
            MyFile.close();
            cout << "Written to file successfully." << endl;
        } else {
            cout << "File is not opened! Error message." << endl;
        }
    }
};

int main() {
    File obj;
    obj.MakeFile();
    return 0;
}
```

Output:-

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
Written to file successfully.
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Lab Task No 14 :-

Input a string into the file.

```
#include <iostream>
#include <fstream>
using namespace std;

class File {
private:
    ofstream MyFile;
public:
    void MakeFile() {
        MyFile.open("file1.txt");
        if (MyFile.is_open()) {
            MyFile << " My name is  Zohaib Khalid ";
            MyFile.close();
            cout << "String written to file successfully.";
        } else {
            cout << "File is not opened! Error message.";
        }
```

```
    }
};

int main() {
    File obj;
    obj.MakeFile();
    return 0;
}
```

Output:-

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
String written to file successfully.
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Lab Task No 15 :-

Output a string from the file.

```
#include <iostream>
#include <fstream>
using namespace std;

class File {
private:
    ifstream MyFile;

public:
    void ReadFile() {
        MyFile.open("file1.txt");
        if (MyFile.is_open()) {
            string line;
            getline(MyFile, line);
            cout << "String from file: " << line << endl;
            MyFile.close();
            cout << "String Reading from the file successfull ";
        }else {
            cout << "File not opened! Error Message" << endl;
        }
    }
};

int main() {
```

```
    File obj;
    obj.ReadFile();
    return 0;
}
```

Output:-

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
String from file:  My name is  Zohaib Khalid
String Reading from the file successfull
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Lab Task No 16 :-

Create a Unary Operator Overloading Program using prefix.

```
#include <iostream>
using namespace std;

class MyNumber {
private:
    int value;

public:
    //MyNumber(int val) : value(val) {}
    MyNumber(int val){
        value = val;
    }

    // Prefix increment operator overloading
    MyNumber operator++() {
        value += 1;
        return *this; // used to return the modified object after performing an
operation
    }

    void display() {
        cout << "Value: " << value << endl;
    }
};

int main() {
    MyNumber num(5);
```

```
    cout << "Real value: ";
    num.display();

    ++num;
    //++num; //changing it will change the number of after profix increament

    cout << "After prefix increment: ";
    num.display();

    return 0;
}
```

Output:-

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
Real value: Value: 5
After prefix increment: Value: 6
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Lab Task No 17 :-

Create a Unary Operator Overloading Program using postfix.

```
#include <iostream>
using namespace std;

class MyNumber {
private:
    int value;

public:
    MyNumber(int val) : value(val) {} //given in the previous code that how this
is working

    // Postfix increment operator overloading
    MyNumber operator++(int) { //int is spesial indicator (dummy parameter).
        MyNumber temp = *this; // Create a copy of the current object
        value += 1; // Increment the value of the current object
        return temp; // Return the copy of the original object
    }

    void display() {
        cout << "Value: " << value << endl;
    }
```

```
};

int main() {
    MyNumber num(5);

    cout << "Real value: ";
    num.display();

    MyNumber result = num++; // Using the overloaded postfix increment operator

    cout << "After postfix increment: ";
    num.display();

    cout << "Value of result: ";
    result.display();

    return 0;
}
```

Output:-

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
Real value: Value: 5
After postfix increment: Value: 6
Value of result: Value: 5
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Lab Task No 18 :-

Add and Subtract using Binary Operator Overloading.

```
#include <iostream>
using namespace std;

class MyNumber {
private:
    int value;

public:
    MyNumber(int val) : value(val) {}

    // Binary addition operator overloading
    MyNumber operator+(const MyNumber& other) {
```

```
        return MyNumber(value + other.value);
    }

    // Binary subtraction operator overloading
    MyNumber operator-(const MyNumber& other) {
        return MyNumber(value - other.value);
    }

    void display() {
        cout << "Value: " << value << endl;
    }
};

int main() {
    MyNumber num1(10);
    MyNumber num2(5);

    cout << "Real values:" << endl;
    num1.display();
    num2.display();

    MyNumber sum = num1 + num2; // Using the overloaded + operator
    MyNumber difference = num1 - num2; // Using the overloaded - operator

    cout << "Sum of values:" << endl;
    sum.display();

    cout << "Difference of values:" << endl;
    difference.display();

    return 0;
}
```

Output:-

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
 if ($?) { .\tempCodeRunnerFile }
Real values:
Value: 10
Value: 5
Sum of values:
Value: 15
Difference of values:
Value: 5
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Lab Task No 19 :-

What is the output of the C++ program when a pointer to the Base class is used to call the display() function on an object of the Derived class, and why does this output occur?

```
/*
When a pointer of base class is used to call the display() function of the
derived class the output
depends on whether the display() function in the Base class is declared as
virtual or not.
*/

//Without virtual (It's base)

#include <iostream>
using namespace std;

class Base {
public:
    void display() {
        cout << "It's of base class display() function. " << endl;
    }
};

class Derived : public Base {
public:
    void display() {
        cout << "It's of derived class display() function. " << endl;
    }
};


// By using Virtual (It's derived)


// class Base {
// public:
//     virtual void display() {
//         cout << "It's of base class display() function" << endl;
//     }
// };

// class Derived : public Base {
// public:
//     void display() {
```

```
//         cout << "It's of derived class display() function" << endl;
//     }
// };


int main() {
    Base* basePtr = new Derived();
    basePtr->display();
    delete basePtr;
    return 0;
}
```

Output:-

Case 1:- (While not using virtual**)**

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
It's of base class display() function.
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Case 2:- (While using virtual**)**

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
It's of derived class display() function
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Lab Task No 20 :-

What is the output of the C++ program when a pointer to the Base class is used to call the display() function on an object of the Derived class, and why does this output occur, considering that the display() function in the Base class is declared as virtual?

```
/*
 The output will be determined by the actual type of the object pointed to at
runtime.
 This is an example of runtime polymorphism and dynamic binding, made possible by
using virtual functions.
*/

#include <iostream>
using namespace std;

class Base {
public:
    virtual void display() {
```

```
        cout << "It's of base class display() function. " << endl;
    }
};

class Derived : public Base {
public:
    void display() {
        cout << "It's of derived class display() function. " << endl;
    }
};

int main() {
    Base* basePtr = new Derived();
    basePtr->display();
    delete basePtr;
    return 0;
}
```

Output:-

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
It's of derived class display() function.
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Lab Task No 21 :-

What will be the output of the C++ program provided, which involves a base class myclass, a derived class derived, and a pointer to the base class pointing to an object of the derived class, considering the virtual function show() in the base class?

```
#include <iostream>
using namespace std;

class myclass {
public:
    virtual void show() {
        cout << "BIt's of base class show() function. " << endl;
    }
};

class derived : public myclass {
public:
    void show() {
        cout << "It's of derived class show() function. " << endl;
```

```
    }
};

int main() {
    myclass* ptr = new derived();
    ptr->show();
    delete ptr;
    return 0;
}
```

Output:-

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
It's of derived class show() function.
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Lab Task No 22 :-

What is the output of the C++ program when an object of the derived class is created and its show() method is called, considering the use of a pure virtual function show() in the base class myclass and the initialization of the protected variable x in the derived class?

```
/*
If a base class contains a pure virtual function, the base class is considered an
abstract
class. An abstract class cannot be instantiated directly; it's meant to be used
as a blueprint
for derived classes.
*/
#include <iostream>
using namespace std;

class myclass {
public:
    virtual void show() = 0; // Pure virtual function
};

class derived : public myclass {
protected:
    int x;

public:
```

```
    derived(int val) : x(val) {}

    void show() {
        cout << "Derived class show(), x = " << x << endl;
    }
};

int main() {
    derived obj(10);
    obj.show();
    return 0;
}
```

Output:-

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
Derived class show(), x = 10
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

Lab Task No 23 :-

What is the output of the C++ program when an object of class3 is created and its inc() method is called, considering the use of virtual inheritance, the inc() method override in various derived classes, and the shared variable x from the base class class1?

```
#include <iostream>
using namespace std;

// Base class
class class1 {
public:
    int x;
    class1() : x(1) {}
};

// Derived class 1 virtually inheriting from class1
class class2 : virtual public class1 {
public:
    void inc() {
        x++; // Increment x by 1
    }
};

// Derived class 2 virtually inheriting from class1
```

```
class class3 : virtual public class1 {
public:
    void inc() {
        x += 2; // Increment x by 2
    }
};

// Derived class inheriting from class2 and class3
class class4 : public class2, public class3 {
public:
    void inc() {
        x += 3; // Increment x by 3
    }
};

int main() {
    class4 obj; // Create an object of class4
    obj.inc();  // Call the inc() method of class4
    cout << "Final value of x: " << obj.x << endl; // Print the final value of x
    return 0;
}
```

Output:-

```
PS D:\2nd-Semester\OOP\OOP Codes> cd "d:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes\"
Final value of x: 4
PS D:\2nd-Semester\OOP\OOP Codes\My Lab Manual all codes>
```

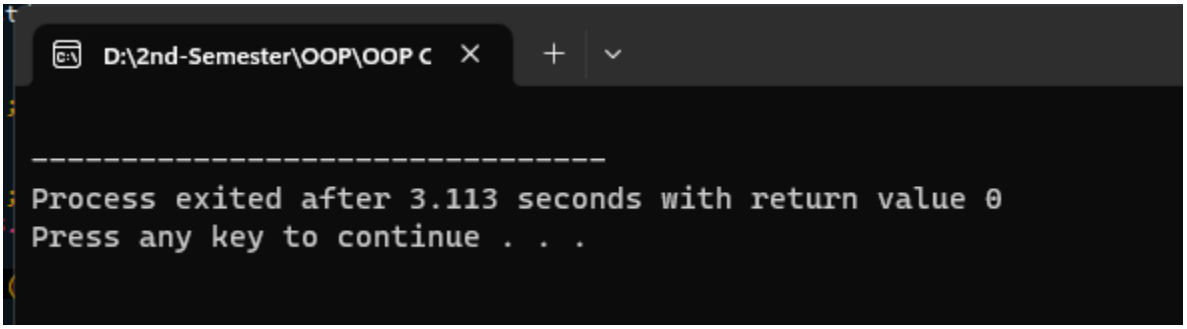
Lab Task No 24 :-

18.1 - Write a program that writes 3 character in a file using formatted I/O .

```
#include <iostream>
#include <stdlib.h>
#include <fstream>
using namespace std;
int main()
{
    int n=10;
    char ch= '*';
    double d= 38.125;
    ofstream file("test.txt");
    if(!file)
    {
```

```
        cout<<"File opening error.";
        exit(1);

    }
    file<<n<<" "<<ch<<" "<<d;
    file.close();

}
```

Output:-

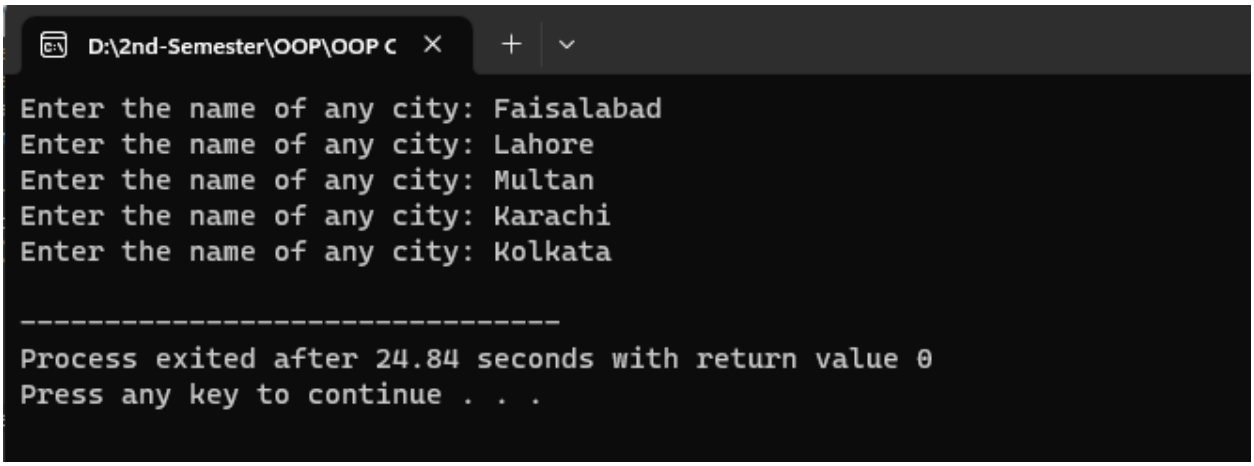


Lab Task No 25 :-

18.2 - Write a program that inputs the names of five cities and store them in a file city.txt

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    char city[50];
    ofstream file("city.txt");
    for(int i=0; i<5; i++)
    {
        cout<<"Enter the name of any city: ";
        cin>>city;
        file<<city<<"\n";
    }
    file.close();
}
```

Output:-

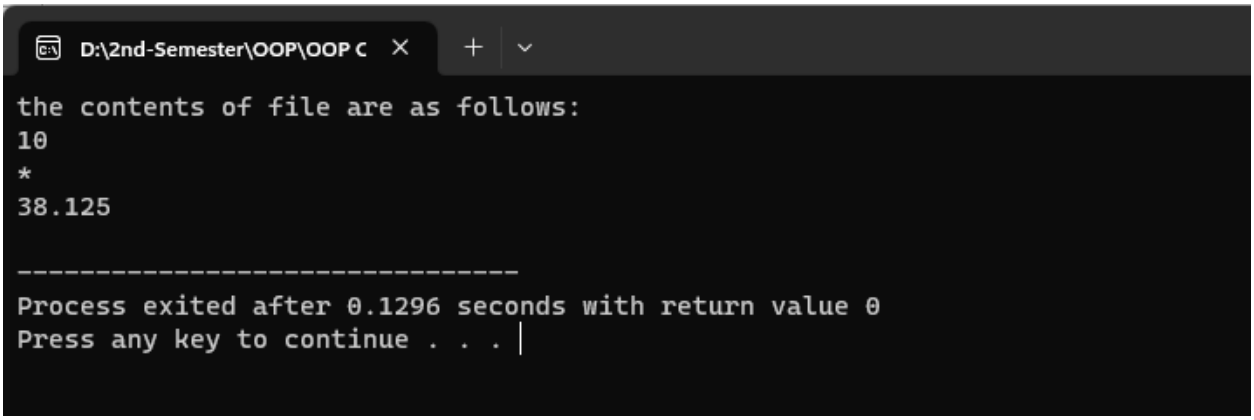


Lab Task No 26 :-

18.3 -- Write a program that reads the contents of the file test.txt and displays on the screen

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    int n;
    char ch;
    double d;
    ifstream file("text.txt");
    if(!file)
    {
        cout<<"File opening error";
        exit(1);
    }
    file>>n>>ch>>d;
    cout<<"the contents of file are as follows: "<<endl;
    cout<<n<<endl<<ch<<endl<<d<<endl;
    file.close();
}
```

Output:-

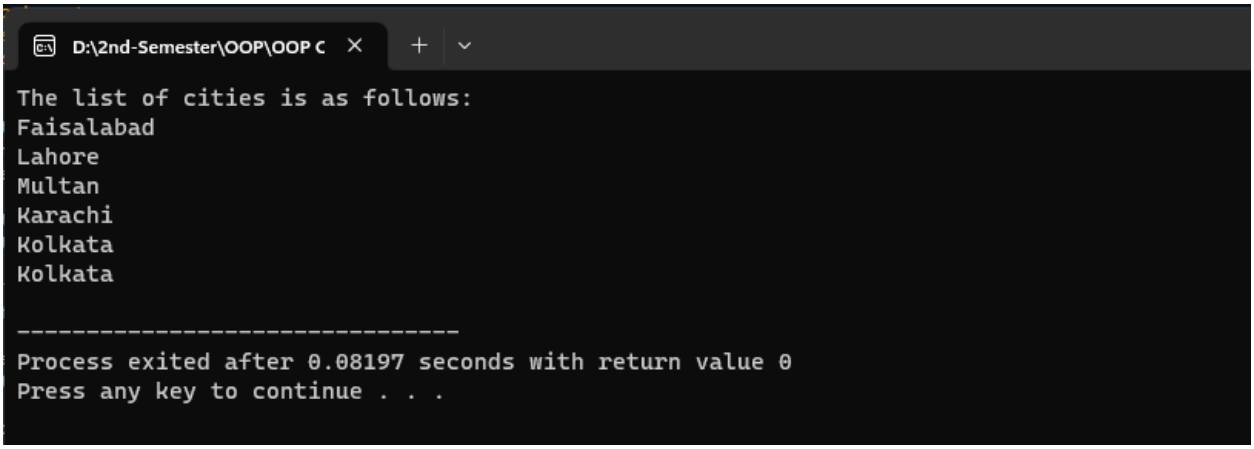


Lab Task No 27 :-

18.4 -- Write a program that displays all records from city.txt prepared in previous example

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    char city[50];
    ifstream file("text.txt");
    if(!file)
    {
        cout<<"Error in opening file";
        exit(1);
    }
    cout<<"The list of cities is as follows: "<<endl;
    while(!file.eof())
    {
        file>>city;
        cout<<city<<endl;
    }
    file.close();
}
```

Output:-



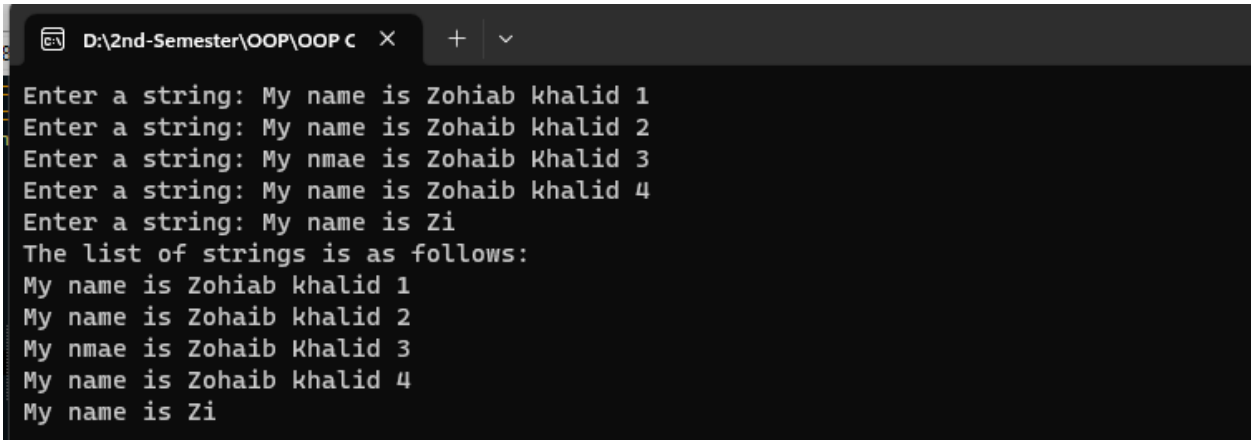
Lab Task No 28 :-

18.5 -- Write a program that stored five lines of a string in a file and then displays them on the screen by reading these lines.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    char str[100];
    ofstream out("strings.txt");
    ifstream in("strings.txt");
    for(int i=0; i<5; i++)
    {
        cout<<"Enter a string: ";
        gets(str);
        out<<str<<"\n";
    }out.close();
    cout<<"The list of strings is as follows: "<<endl;
    while(!in.eof())
    {
```

```
        in.getline(str, 100);
        cout<<str<<endl;
    }
    in.close();
}
```

Output:-

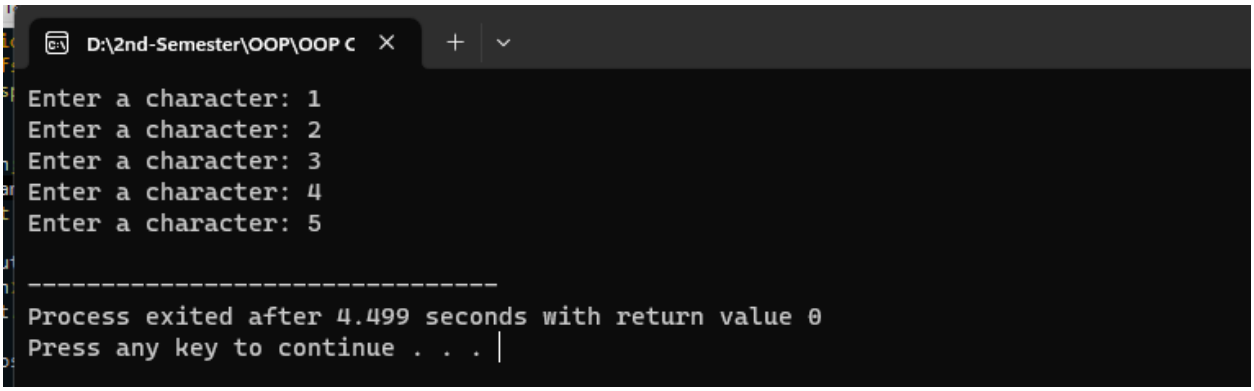


Lab Task No 29 :-

18.6 -- Write a program that inputs five characters from the user and stores them in a file

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    char ch;
    ofstream out("chars.txt");
    for(int i=0; i<5; i++)
    {
        cout<<"Enter a character: ";
        cin>>ch;
        out.put(ch);
    }
    out.close();
}
```

Output:-

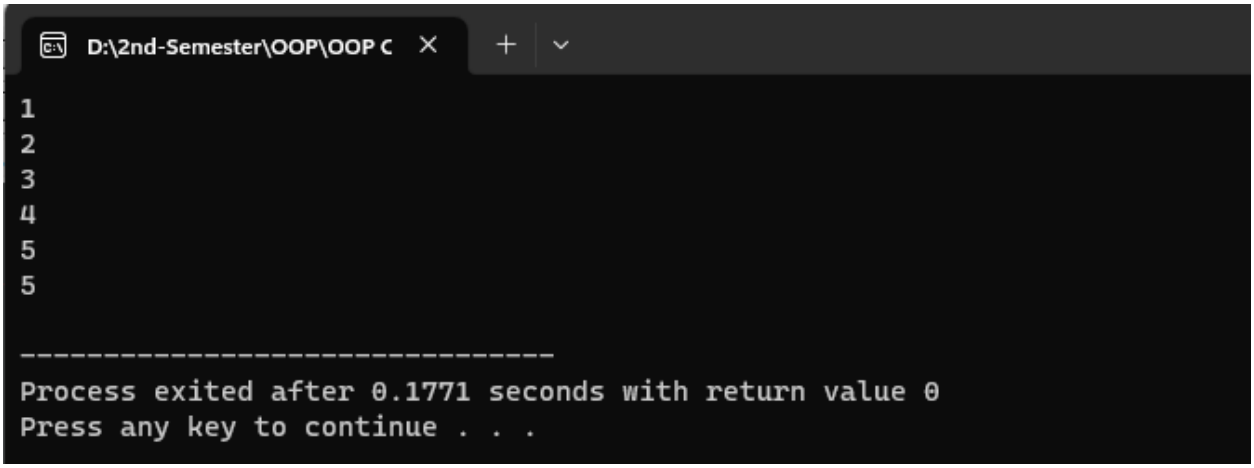


Lab Task No 30 :-

18.7 -- Write a program that writes a characters from a text file and displays them on screen

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    char ch;
    ifstream in("chars.txt");
    while(!in.eof())
    {
        in.get(ch);
        cout<<ch<<endl;
    }
    in.close();
}
```

Output:-

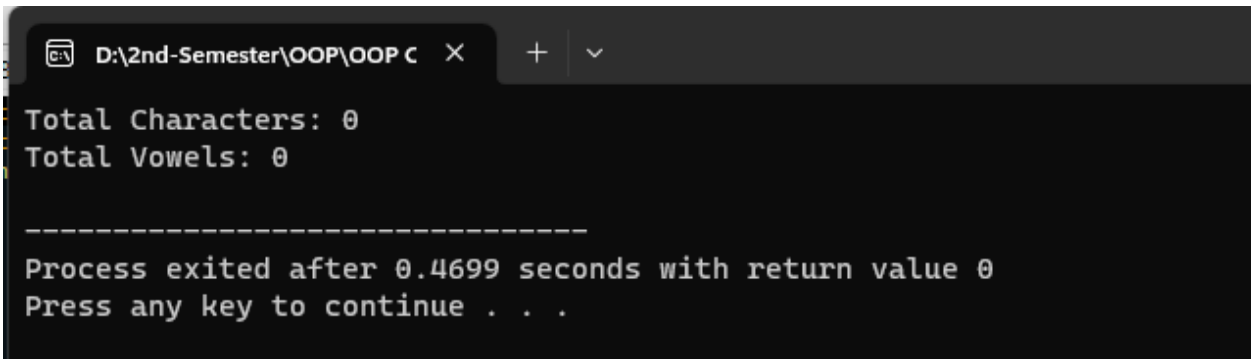


Lab Task No 31 :-

18.8 -- Write a program that reads the character from a text file. It counts total number of characters and total number of vowels in the file.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    char ch;
    int t, v;
    t=v=0;
    ifstream in("chars.txt");
    while(!in.eof())
    {
        in.get(ch);
        ch=tolower(ch);
        if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
        {
            v++;
            t++;
            cout<<ch<<endl;
        }
    }
    cout<<"Total Characters: "<<t<<endl;
    cout<<"Total Vowels: "<<v<<endl;
    in.close();
    return 0;
}
```

Output:-

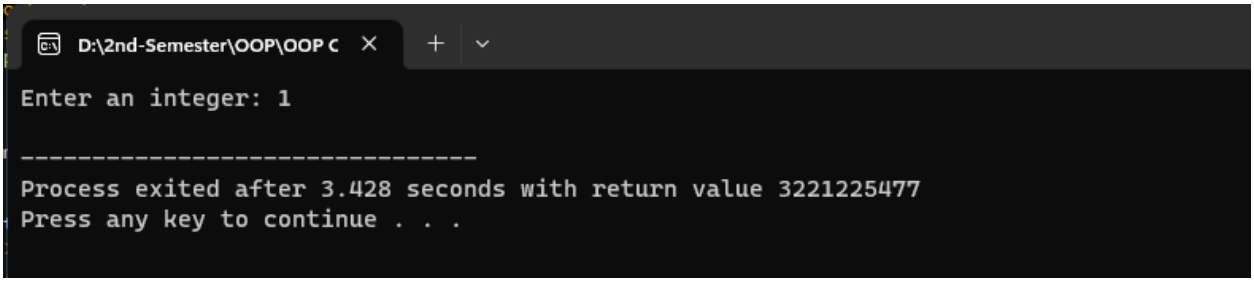


Lab Task No 32 :-

18.9 -- Write a program that inputs five integers and stores it in a file using binary I/O.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    int n;
    ofstream out("data.txt" , ios::binary);
    for(int i=1; i<=5; i++)
    {
        cout<<"Enter an integer: ";
        cin>>n;
        out.write((char*)n, sizeof(n));
    }
    out.close();
}
```

Output:-



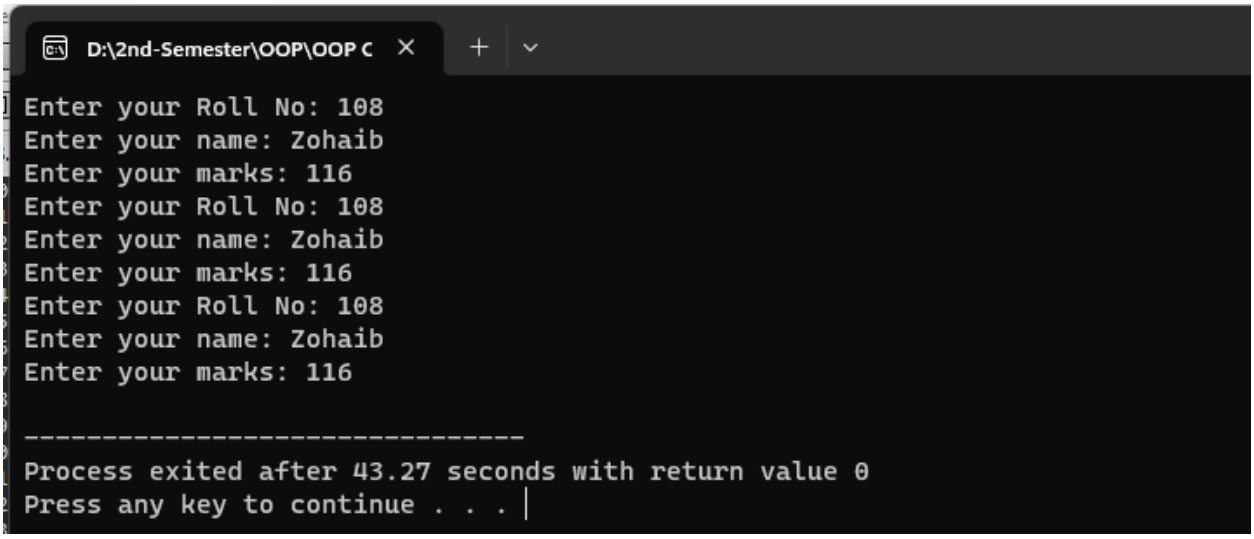
Lab Task No 33 :-

18.10 -- Write a program that uses a structure variable to input records of three students and stores it in students.txt in binary I/O. Each record console of Roll No, Name and Marks

```
#include <iostream>
#include <fstream>
using namespace std;
struct Student
{
    int rno;
    char name[50];
    int marks;
};
int main()
```

```
{
    Student s;
    ofstream out("student.txt", ios::binary);
    for(int i=1; i<=3; i++)
    {
        cout<<"Enter your roll no: ";
        cin>>s.rno;
        cout<<"Enter your name: ";
        cin>>s.name;
        cout<<"Enter your marks: ";
        cin>>s.marks;
        out>write((char*)&s, sizeof(s));
    }
    out.close();
}
```

Output:-



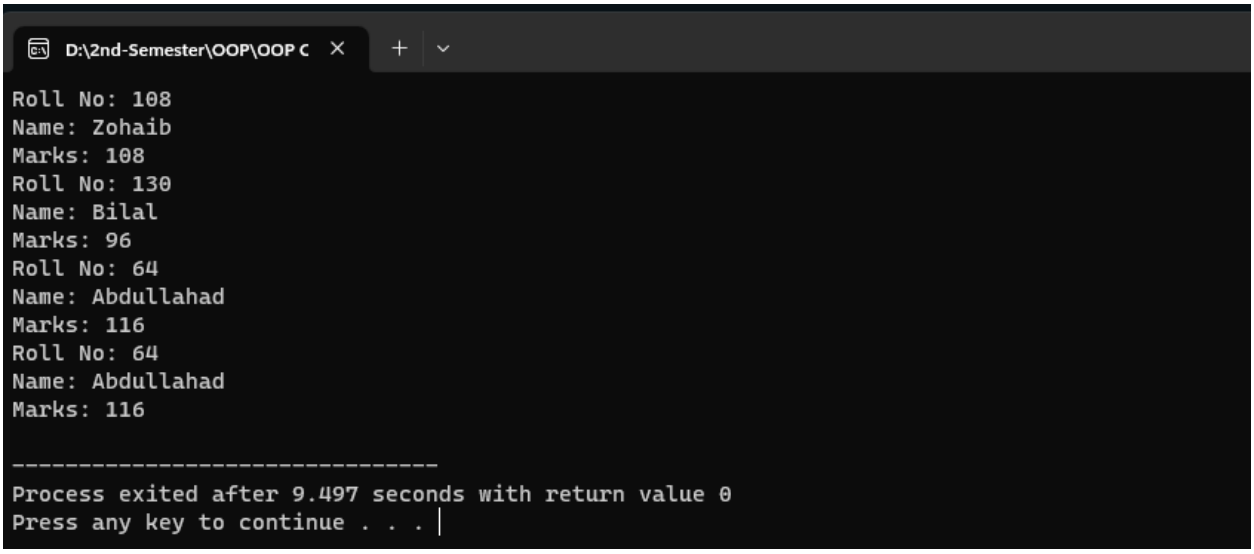
Lab Task No 34 :-

18.11 -- Write a program that reads the records stored in students.txt file and displays them

```
#include <iostream>
#include <fstream>
using namespace std;
struct Student
{
    int rno;
    char name[50];
```

```
    int marks;
};
int main()
{
    Student s;
    ifstream in("students.txt", ios::binary);
    while(!in.eof())
    {
        in.read((char*)&s, sizeof(s));
        cout<<"Roll No: "<<s.rno<<endl;
        cout<<"Name: "<<s.name<<endl;
        cout<<"Marks: "<<s.marks<<endl;
    }
    in.close();
}
```

Output:-



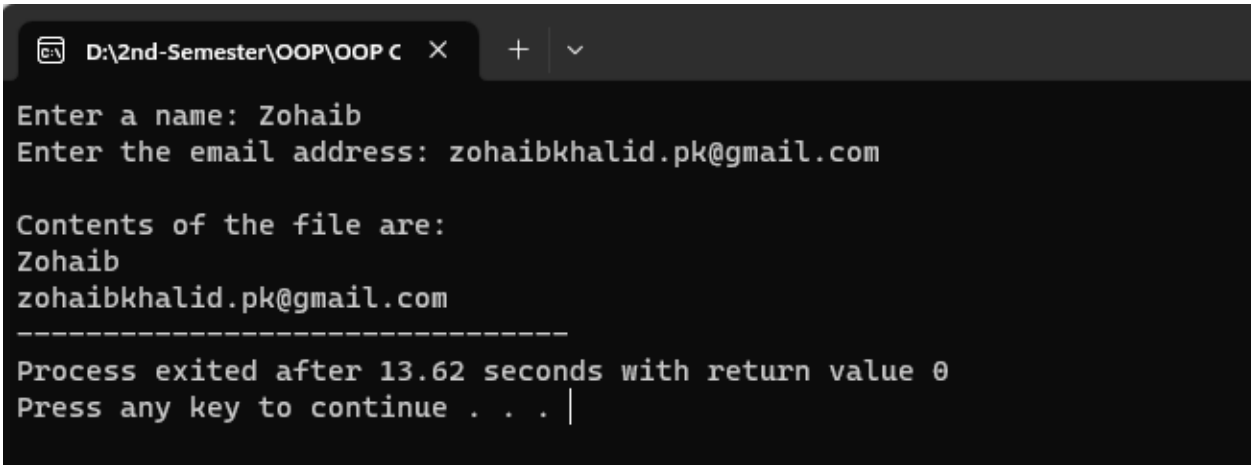
Lab Task No 35 :-

18.12 -- Write a program that creates a file to store name and email of the user using structure.

```
#include <iostream>
#include <fstream>
using namespace std;
struct email
```

```
{
    char name[20];
    char id[30];
};
int main()
{
    email user;
    email check;
    cout<<"Enter a name: ";
    cin>>user.name;
    cout<<"Enter the email address: ";
    cin>>user.id;
    ofstream out("email.txt", ios::out | ios::binary);
    out.write((char*)&user, sizeof(struct email));
    out.close();
    cout<<endl<<"Contents of the file are: ";
    ifstream in ("email.txt", ios::in | ios::binary);
    in.read((char*)&check, sizeof(struct email));
    cout<<endl<<check.name;
    cout<<endl<<check.id;
    in.close();
};
```

Output:-



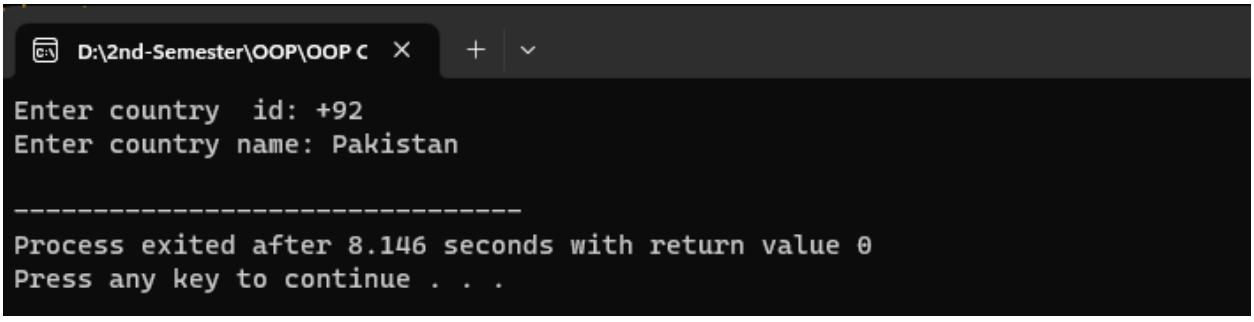
Lab Task No 36 :-

18.13 -- Write a program that stores an object to a file country.txt using binary I/O.

```
#include <iostream>
```

```cpp
#include <fstream>
using namespace std;
class country
{
    private:
        int id;
        char name[50];
    public:
        void get()
        {
            cout<<"Enter country  id: ";
            cin>>id;
            cout<<"Enter country name: ";
            cin>>name;
        }

        void show()
        {
            cout<<"Country Id: "<<id<<endl;
            cout<<"Country Name: "<<name<<endl;
        }
};
int main()
{
    country cn;
    ofstream out("country.txt, ios::binary");
    cn.get();
    out.write((char*)&cn, sizeof(cn));
    out.close();
}
```

Output:-



Lab Task No 37 :-

18.14 -- Write a program that reads the contents of country.txt and display on the screen.

```
#include <iostream>
#include <fstream>
using namespace std;
class Country
{
    private:
        int id;
        char name[50];
        public:
            void get()
            {
                cout<<"Enter country id: ";
                cin>>id;
                cout<<"Enter country name: ";
                cin>>name;
            }

            void show()
            {
                cout<<"Country Id: "<<id<<endl;
                cout<<"Country Name: "<<name<<endl;
            }
};
int main()
{
    Country cn;
    ifstream in("country.txt", ios::binary);
    in.read((char*)&cn, sizeof(cn));
    cn.show();
    in.close();
}
```

Output:-