

Department: COMPUTER SCIENCE AND ENGINEERING

Semester: Fall 2023

Program: Bachelor of Computer Science and Engineering

Course Title: Object Oriented Programming - I

Course Code : 0613-CSE-1209

**ASSIGNMENT 01**

|  |  |
| --- | --- |
| **Student Name:** | Md. Siam Hossain |
| **Student ID:** | 23100084 |
| **Student Batch:** | 26th |
| **Submission Date:** | 02/12/2023 |
|  |  |
| **Course Teacher:** | **Ankan Ray**  **Lecturer, CSE, RPSU** |

**NARAYANGANJ 2023**

///Topics-01: Virtual Function

* Source Code:

#include <iostream>

#include <conio.h>

using namespace std;

// Using Abstract base class defining the Virtual Function for car drivers

class carDriver

{

public:

    virtual void information() = 0; // Pure virtual function for displaying information

    virtual void get() = 0;         // Pure virtual function for getting input

};

// Derived class Siam implementing the carDriver interface

class Siam : public carDriver

{

public:

    int id;

    int salary;

    // Implementation of the get function for Siam

    void get()

    {

        cout << "Enter the ID and Salary of Siam: ";

        cin >> id >> salary;

    }

    // Implementation of the information function for Siam

    void information()

    {

        cout << "The ID No of Siam is: " << id << endl;

        cout << "The Monthly salary of Siam is: " << salary << endl;

    }

};

// Derived class Talha implementing the carDriver interface

class Talha : public carDriver

{

public:

    int id;

    string home\_add;

    // Implementation of the get function for Talha

    void get()

    {

        cout << endl << "Enter the ID and Home Address of Talha: ";

        cin >> id;

        getline(cin, home\_add);

    }

    // Implementation of the information function for Talha

    void information()

    {

        cout << "The ID No of Talha is: " << id << endl;

        cout << "The Home Address of Talha is: " << home\_add << endl;

    }

};

int main()

{

    carDriver \*v;

    Siam s;

    Talha t;

    // Using a pointer to the base class of accessing Siam class properties

    v = &s;

    v->get();

    v->information();

    // Using a pointer to the base class of accessing Talha class properties

    v = &t;

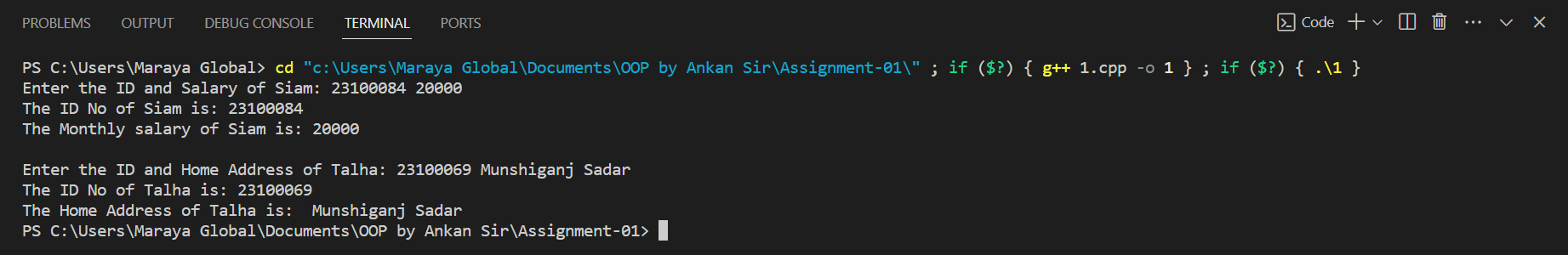
    v->get();

    v->information();

    return 0;

}

* Input and Output:



///Topics-2.1: Using arrays of objects and references of objects.

* Source Code:

#include<iostream>

using namespace std;

class worker

{

    public:

    int id;

    int age;

    void set(int i)

    {

      cout<< "Enter the ID and Age of the "<< i+1 << "th Workes's: ";

      cin>> id >> age;

    }

    void display(int i)

    {

      cout<< "The ID No of "<< i+1 << "th Workers is: "<< id << endl;

      cout<< "The Age of "<< i+1 << "th Workers is: "<< age << endl;

    }

};

int main()

{

    worker person[3];

    for(int i=0;i<3;i++)

    {

        person[i].set(i);

        person[i].display(i);

    }

    // Using Refrence of Objects

    worker &ref\_1 = person[0];

    worker &ref\_2 = person[1];

    ref\_1.id = 23100069;

    ref\_1.age = 25;

    ref\_2.id = 2310169;

    ref\_2.age = 30;

    for(int i=0;i<3;i++)

    {

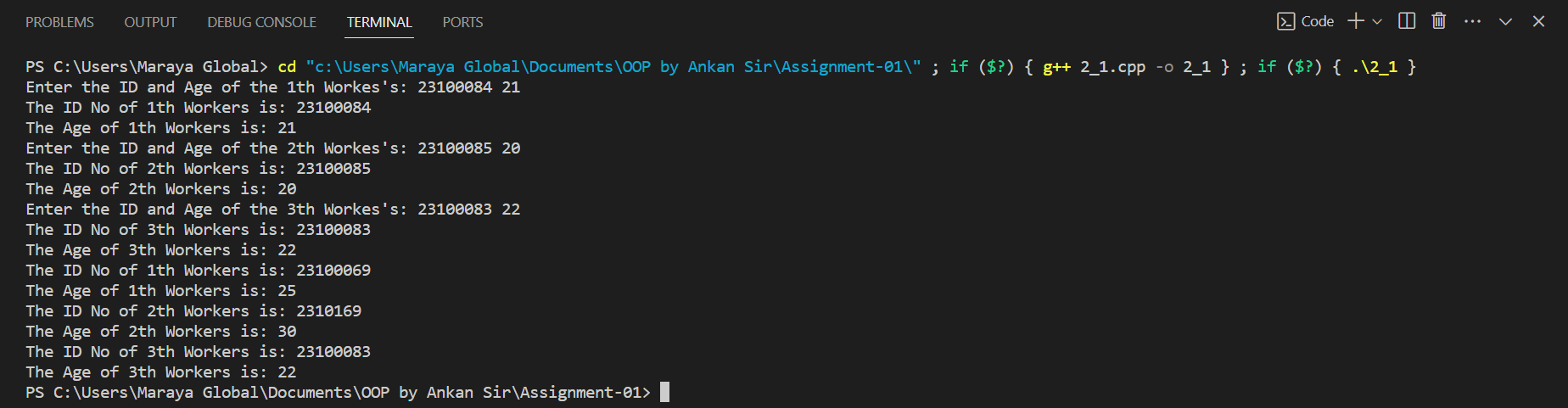
        person[i].display(i);

    }

    return 0;

}

* Input and Output:



///Topics-2.2: Using objects as arguments and returning objects from functions

* Source Code:

#include <iostream>

using namespace std;

class student

{

private:

    int marks;

    int total\_marks;

public:

    // Function to set the marks for a student

    void set\_marks(int i)

    {

        cout << "Enter the marks of "<< i << "th student is: ";

        cin >> marks;

    }

    // Function to add the marks of two students

    void add\_marks(student m1, student m2)

    {

        total\_marks = m1.marks + m2.marks;

    }

    // Function to display the total marks

    void display()

    {

        cout << "Total Marks is: " << total\_marks;

    }

};

int main()

{

    // Creating three student objects

    student s1, s2, s3;

    // Setting marks for the first two students

    s1.set\_marks(1);

    s2.set\_marks(2);

    // Adding the marks of the first two students and storing the result in the third student

    s3.add\_marks(s1, s2);

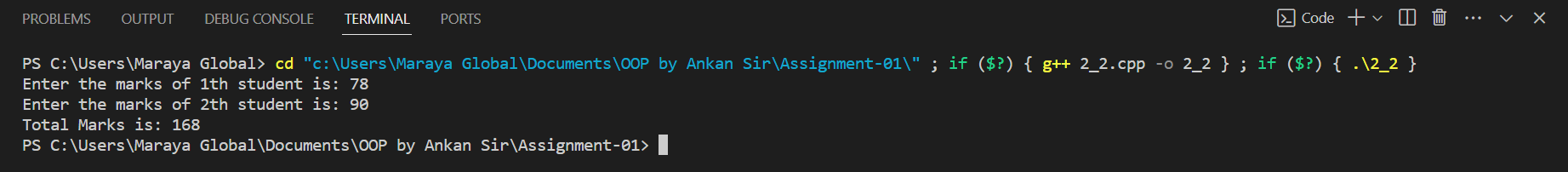
    // Displaying the total marks of the third student

    s3.display();

    return 0;

}

* Input and Output:



///Topics-3.1: Template functions

* Source Code:

#include<iostream>

using namespace std;

// Template function 'ope' that takes two parameters of type 'operation'

// and returns the result of the addition of those parameters.

template<class operation>

operation ope(operation a, operation b)

{

    return a + b;

}

int main()

{

    // Using the 'ope' function with integer parameters 5 and 7.

    cout << "The Addition of 5 and 7 is: "<< ope(5, 7) << endl;

    // Using the 'ope' function with floating-point parameters 5.5 and 6.7.

    cout << "The Addition of 5.5 and 6.7 is: "<< ope(5.5, 6.7) << endl;

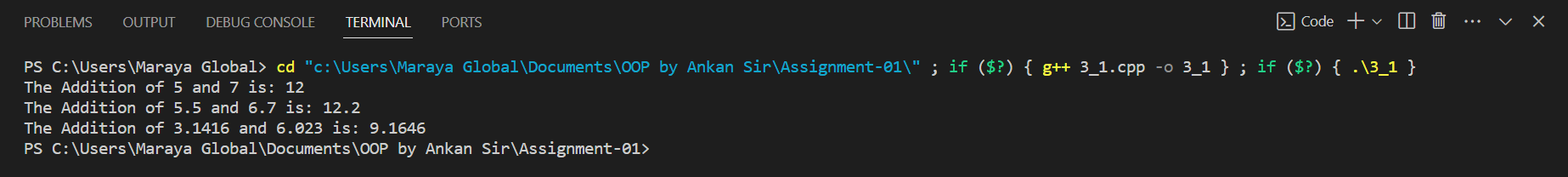
    // Using the 'ope' function with floating-point parameters 3.1416 and 6.023.

    cout << "The Addition of 3.1416 and 6.023 is: "<< ope(3.1416, 6.023) << endl;

    return 0;

}

* Input and Output:



///Topics-3.2: Template Classes

* Source Code:

#include <bits/stdc++.h>

using namespace std;

// Class template definition for oop

template <typename T>

class oop

{

private:

    T first; // Private member variable

    T sec;

public:

    // Constructor to initialize the object with two values

    oop(T a, T b)

    {

        first = a;

        sec = b;

    }

    // Member function to perform addition

    void addition();

    // Member function to perform multiplication

    void multiplication();

};

// Definition of the addition member function

template <typename T>

void oop<T>::addition()

{

    cout << "Addition of "<< first <<" and "<< sec << " is: " << first + sec << endl;

}

// Definition of the multiplication member function

template <typename T>

void oop<T>::multiplication()

{

    cout << "Multiplication of "<< first <<" and "<< sec << " is: " << first \* sec << endl;

    cout << endl;

}

int main()

{

    // Create an object of oop class with int data type

    oop<int> int\_num(10, 15);

    int\_num.addition();

    int\_num.multiplication();

    // Create an object of oop class with double data type

    oop<double> double\_num(3.1416, 6.023);

    double\_num.addition();

    double\_num.multiplication();

    return 0;

}

* Input and Output:

