ASSIGNMENT

COURSE TITLE	:	Object Oriented Programming
COURSE CODE	:	CSE 111-112
ASSIGNMENT NO.	:	02
SUBMISSION DATE	:	11-12-2024

SUBMITTED TO

NAME	:	Ashfia Jannat Keya
DEPT. OF	:	Computer Science and Engineering (CSE) Bangladesh University of Business & Technology (BUBT)

SUBMITTED BY

NAME	:	Shadman Shahriar
ID NO.	:	20245103408
INTAKE	:	53
SECTION	:	4
PROGRAM	:	B.Sc. Engg. in CSE

Object Oriented Programming Lab Report 1 (Section 4)

1. `Company` Class

Create a `Company` class with attributes called `no_of_employee`, `no_of_dept`, `employee_id`, `employee_name` and a function `set_value()` to initialize the attributes. Create another class HR with attributes, address, salary, contact and a function `set_value()` to set the values. Now create an `Employee` class which inherits `Company` class and `HR` class. Now call the functions of the parent class with the child's object.

Ans. Here is a C++ program that satisfies the conditions above:

```
/**
 * ======================
 * Name: Shadman Shahriar
 * ID  : 20245103408
 * ======================
 */

#include <iostream>
using namespace std;

class Company
{
protected:
    int no_of_employee;
    int no_of_dept;
    int employee_id;
    string employee_name;
```

```
public:
    void set_value(int nE, int nD, int eID, string eName)
    {
        no_of_employee = nE;
        no_of_dept = nD;
        employee_id = eID;
        employee_name = eName;
    }
};

class HR
{
protected:
    string address;
    int salary;
    string contact;

public:
    void set_value(string a, int s, string c)
    {
        address = a;
        salary = s;
        contact = c;
    }
};

class Employee : public Company, public HR
{
public:
    void display()
    {
        cout << "No. of Employees: " << no_of_employee <<
endl;
        cout << "No. of Depts.    : " << no_of_dept << endl;
        cout << "Employee ID      : " << employee_id <<
endl;
        cout << "Employee Name    : " << employee_name <<
endl;
        cout << "Address         : " << address << endl;
        cout << "Salary          : " << salary << endl;
        cout << "Contact         : " << contact << endl;
    }
};

int main()
{
    Employee kamla;
    kamla.Company::set_value(154, 20, 408, "S. Shahriar");
    kamla.HR::set_value("Road no. 32, Rupnagar R/A, Mirpur
- 1216", 60000, "+88018XXXXXXX3");
    kamla.display();

    return 0;
}
```

Output: The code yields the following output in the terminal:

```
No. of Employees: 154
No. of Depts.   : 20
Employee ID     : 408
Employee Name   : S. Shahriar
Address         : Road no. 32, Rupnagar R/A, Mirpur - 1216
Salary          : 60000
Contact         : +88018XXXXXXX3
```

2. `Shape` Class

Create a class named `Shape` with a function that prints **"This is a shape"**. Create another class named Polygon inheriting the `Shape` class with the same function that prints **"Polygon is a shape"**. Create two other classes named `Rectangle` and `Triangle` having the same function which prints **"Rectangle is a polygon"** and **"Triangle is a polygon"** respectively. Again, make another class named `Square` having the same function which prints **"Square is a rectangle"**. Now, try calling the function by the object of each of these classes.

Ans. Here is a C++ program that satisfies the conditions above:

```cpp
/**
 * ======================
 * Name: Shadman Shahriar
 * ID  : 20245103408
 * ======================
 */

#include <iostream>
using namespace std;

class Shape
{
public:
    void display()
    {
        cout << "This is a shape." << endl;
    }
};

class Polygon : public Shape
{
public:
    void display()
    {
        cout << "Polygon is a shape." << endl;
    }
};

class Rectangle : public Polygon
{
public:
    void display()
    {
        cout << "Rectangle is a polygon." << endl;
    }
};

class Triangle : public Polygon
{
public:
    void display()
    {
        cout << "Triangle is a polygon." << endl;
    }
};

class Square : public Rectangle
{
public:
    void display()
    {
        cout << "Square is a rectangle." << endl;
    }
};

int main()
{
    Shape shape;
    Polygon polygon;
    Rectangle rectangle;
    Triangle triangle;
    Square square;

    shape.display();
    polygon.display();
    rectangle.display();
    triangle.display();
    square.display();
    return 0;
}
```

Output: The code yields the following output in the terminal:

```
This is a shape.
Polygon is a shape.
Rectangle is a polygon.
Triangle is a polygon.
Square is a rectangle.
```

3. `Marks` Class

We want to calculate the total marks of each student of a class in `Physics`, `Chemistry` and `Mathematics` and the average marks of the class. The number of students in the class are entered by the user. Create a class named `Marks` with data members for `roll number`, `name` and `marks`. Create three other classes inheriting the `Marks` class, namely **Physics**, **Chemistry** and **Mathematics**, which are used to define marks in individual subjects of each student. Roll number of each student will be generated automatically.

Ans. Here is a C++ program that satisfies the conditions above:

```cpp
/**
 * ======================
 * Name: Shadman Shahriar
 * ID  : 20245103408
 * ======================
 */

#include <iostream>
using namespace std;

class Marks
{
protected:
    string name;
    int roll;
    int marks;

    Marks(string n, int r, int m)
    {
        name = n;
        roll = r;
        marks = m;
    }
```

```
public:
    float get()
    {
        return marks;
    }
};

class Physics : public Marks
{
public:
    Physics(string n, int r, int m) : Marks(n, r, m) {}
};

class Chemistry : public Marks
{
public:
    Chemistry(string n, int r, int m) : Marks(n, r, m) {}
};

class Mathematics : public Marks
{
public:
    Mathematics(string n, int r, int m) : Marks(n, r, m) {}
};

int main()
{
    int rollOffset = 100;
    int count = 0;
    float total = 0;
    float avg = 0.00;
    cout << "Enter the numbers of students: ";
    cin >> count;

    for (int i = 1; i <= count; i++)
    {
        string name;
        int roll = i + rollOffset;
        int marksP, marksC, marksM, marks;

        cout << "Student " << i << ": " << endl;
        cout << "Name: ";
        cin >> name;
        cout << "Roll: " << roll << endl;
        cout << "Marks obtained in Physics    : ";
        cin >> marksP;
        cout << "Marks obtained in Chemistry  : ";
        cin >> marksC;
        cout << "Marks obtained in Mathematics: ";
        cin >> marksM;

        Physics subjectPhysics(name, roll, marksP);
        Chemistry subjectChemistry(name, roll, marksC);
        Mathematics subjectMathematics(name, roll, marksM);

        marks = subjectPhysics.get() + subjectChemistry.get()
+ subjectMathematics.get();

        cout << "Total marks: " << marks << endl;
        cout << endl;
        total += (float)marks;
    }

    avg = (total / count);
    cout << "Average marks of " << count << " student(s): "
<< avg << endl;
    return 0;
}
```

Output: The code yields the following output in the terminal:

```
+ Enter the numbers of students: 4
  Student 1:
+ Name: S1
```

```
  Roll: 101
+ Marks obtained in Physics    : 55
+ Marks obtained in Chemistry  : 53
+ Marks obtained in Mathematics: 76
  Total marks: 184

  Student 2:
+ Name: S2
  Roll: 102
+ Marks obtained in Physics    : 87
+ Marks obtained in Chemistry  : 86
+ Marks obtained in Mathematics: 55
  Total marks: 228

  Student 3:
+ Name: S3
  Roll: 103
+ Marks obtained in Physics    : 96
+ Marks obtained in Chemistry  : 99
+ Marks obtained in Mathematics: 91
  Total marks: 286

  Student 4:
+ Name: S4
  Roll: 104
+ Marks obtained in Physics    : 96
+ Marks obtained in Chemistry  : 95
+ Marks obtained in Mathematics: 89
  Total marks: 280

  Average marks of 4 student(s): 244.5
```

4. `Library` Class

Create a class named `Library` with attributes such as `lib_name` and `location`. Add a function `set_library()` to initialize the attributes. Derive a class `Section` that inherits the `Library` class, with attributes `section_name` and `section_code`. Add a function `set_section()` to initialize its attributes. Further, derive another class `Book` from the `Section` class, with attributes `book_name`, `author`, and `isbn`. Add a function `display_info()` to display all details about a book. Implement the concept of multilevel inheritance and display the attributes using the most derived class's object.

Ans. Here is a C++ program that satisfies the conditions above:

```cpp
/**
 * ======================
 * Name: Shadman Shahriar
 * ID  : 20245103408
 * ======================
 */

#include <iostream>
using namespace std;

class Library
{
protected:
    string lib_name, location;

public:
    void set_library(string lN, string ln)
    {
        lib_name = lN;
        location = ln;
    }
```

```
};

class Section : public Library
{
protected:
    string section_name;
    int section_code;

public:
    void set_section(string sn, int sc)
    {
        section_name = sn;
        section_code = sc;
    }
};

class Book : public Section
{
protected:
    string book_name, author;
    long long int isbn;

public:
    Book(string bn, string a, long long int i) : book_name(bn
author(a), isbn(i) {}

    void display_info()
    {
        cout << "Library Name: " << Library::lib_name <<
endl;
        cout << "Location: " << Library::location << endl;
        cout << endl;
        cout << "Section Name: " << Section::section_name
<< endl;
        cout << "Section Code: " << Section::section_code
<< endl;
        cout << endl;
        cout << "Book Information:" << endl;
        cout << "Title      : " << book_name << endl;
        cout << "Author     : " << author << endl;
        cout << "ISBN Number: " << isbn << endl;
    }
};

int main()
{
    Book book("Never Give Up", "Shahriar", 1100234532);
    book.set_library("Hriddy", "Mirpur 11");
    book.set_section("Self Help", 154);
    book.display_info();
    return 0;
}
```

Output: The code yields the following output in the terminal:

```
Library Name: Hriddy
Location: Mirpur 11

Section Name: Self Help
Section Code: 154

Book Information:
Title      : Never Give Up
Author     : Shahriar
ISBN Number: 1100234532
```

5. `Department` Class

Create a class `Department` with attributes `dept_name` and `dept_code`. Add a function `set_dept()` to initialize the attributes. Create a separate class Professor with attributes `prof_name` and `prof_id`. Add a function `assign_department()` to assign a department to a professor. Derive a class `HOD` (Head of Department) from Professor, which includes additional attributes `hod_start_date` and `hod_end_date`. Implement the functionality to display all details.

Ans. Here is a C++ program that satisfies the conditions above:

```cpp
/**
 * ======================
 * Name: Shadman Shahriar
 * ID  : 20245103408
 * ======================
 */

#include <iostream>
using namespace std;

class Department
{
protected:
    string dept_name;
    int dept_code;

public:
    void set_dept(string dN, int dC)
    {
        dept_name = dN;
        dept_code = dC;
    }
    string get_dept()
    {
        return dept_name;
    }

    int get_dept_code()
    {
        return dept_code;
    }
};

class Professor
{
protected:
    string prof_name;
    int prof_id;
    Department prof_dept;

public:
    Professor(string n, int i)
    {
        prof_name = n;
        prof_id = i;
    }

    void assign_department(Department d)
    {
        prof_dept.set_dept(d.get_dept(),
d.get_dept_code());
    }
};

class HOD : public Professor
{
protected:
    int hod_start_date;
    int hod_end_date;
```

```
public:
    HOD(string n, int i, int hsd, int hed) : Professor(n,
i), hod_start_date(hsd), hod_end_date(hed) {}

    void display_info()
    {
        cout << "Department    : " << prof_dept.get_dept()
<< " (" << prof_dept.get_dept_code() << ")" << endl;
        cout << "Professor Name: " << prof_name << endl;
        cout << "Professor ID  : " << prof_id << endl;
        cout << "HOD Start Date: " << hod_start_date <<
endl;
        cout << "HOD End Date  : " << hod_end_date << endl;
    }
};

int main()
{
    Department department;
    department.set_dept("Computer Science and Engineering",
101);

    HOD professor("Shahriar", 408, 1025, 1225);
    professor.assign_department(department);
    professor.display_info();
    return 0;
}
```

Output: The code yields the following output in the terminal:

```
Department    : Computer Science and Engineering (101)
Professor Name: Shahriar
Professor ID  : 408
HOD Start Date: 1025
HOD End Date  : 1225
```