Object Oriented Programming Lab

Spring2025

Assignment #10

Umar Farooq
09-131242-088
BSE-2B

DEPARTMENT OF SOFTWARE ENGINEERING
BAHRIA UNIVERSITY ISLAMABAD CAMPUS

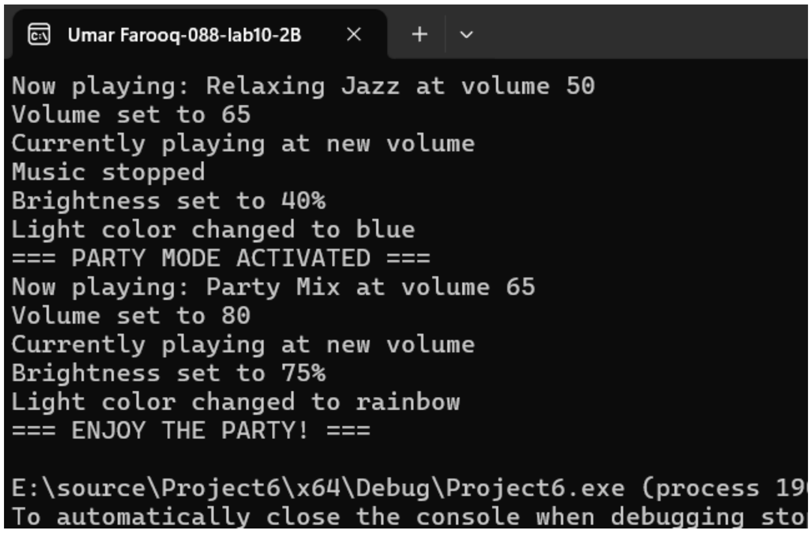
Task 01: Write a program to model a smart home device that combines features of a Speaker (play music, volume control) and a Light (adjust brightness, change color). Create a base class Speaker with methods play_music() and set_volume(), and another base class Light with methods adjust_brightness() and change_color(). Derive a class SmartLamp from both Speaker and Light, adding a new method party_mode() that activates both music and colored lights. The program should allow users to control all features of the SmartLamp without modifying any base class methods. The output should demonstrate the combined functionalities.

Code:

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

class Speaker {
public:
   string song;
   int volume;
   bool isPlaying;

   Speaker()
   {
      song = "Relaxing Jazz";
      volume = 50;
      isPlaying = false;
   }

   void play_music()
   {
      isPlaying = true;
      cout << "Now playing: " << song << " at volume " << volume << endl;
   }

   void set_volume(int newVolume)
   {
      volume = newVolume;
      cout << "Volume set to " << volume << endl;
      if (isPlaying)
      {
         cout << "Currently playing at new volume" << endl;
```

```
        }
    }

    void stop_music()
    {
        isPlaying = false;
        cout << "Music stopped" << endl;
    }
};

class Light {
public:
    int brightness;
    string color;

    Light()
    {
        brightness = 100;
        color = "White";
    }

    void adjust_brightness(int level)
    {
        brightness = level;
        cout << "Brightness set to " << brightness << "%" << endl;
    }

    void change_color(const string& newColor)
    {
        color = newColor;
        cout << "Light color changed to " << color << endl;
    }
};

class SmartLamp : public Speaker, public Light {
public:
    void party_mode()
    {
        cout << "=== PARTY MODE ACTIVATED ===" << endl;
        song = "Party Mix";
        volume = 65;
        play_music();
        set_volume(80);
        adjust_brightness(75);
        change_color("rainbow");
        cout << "=== ENJOY THE PARTY! ===" << endl;
    }
};

int main() {
    SmartLamp l;

    l.play_music();
    l.set_volume(65);
    l.stop_music();
    l.adjust_brightness(40);
    l.change_color("blue");

    l.party_mode();

    return 0;
}
```

Screenshot:



```
Now playing: Relaxing Jazz at volume 50
Volume set to 65
Currently playing at new volume
Music stopped
Brightness set to 40%
Light color changed to blue
=== PARTY MODE ACTIVATED ===
Now playing: Party Mix at volume 65
Volume set to 80
Currently playing at new volume
Brightness set to 75%
Light color changed to rainbow
=== ENJOY THE PARTY! ===

E:\source\Project6\x64\Debug\Project6.exe (process 19
To automatically close the console when debugging sto
```

Task 03: Model a hospital staff system where a Surgeon is both a Doctor (specialization, license number) and an Administrator (department, management level). Create a base class HospitalStaff with name, address(string) and staff_id. Using virtual inheritance, derive Doctor and Administrator from HospitalStaff, then create a Surgeon class inheriting from both. The program should ensure HospitalStaff attributes are not duplicated and display all surgeon details.

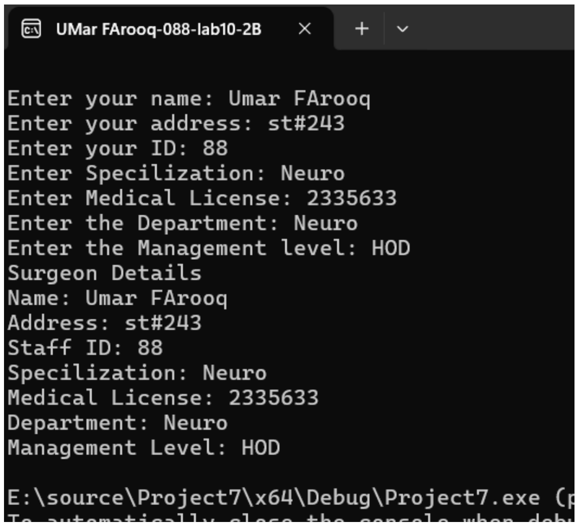
Code:

```
#include<iostream>
#include<string>
using namespace std;
class HospitalStaff {
protected:
        string name;
        string address;
        int staff_id;
public:
        HospitalStaff()
        {
                name = " ";
                address = " ";
                staff_id = 0;
        }
        HospitalStaff(string n, string add, int id)
        {
                name = n;
                address = add;
                staff_id = id;
        }
        void setinput()
        {
                cin.ignore();
                cout << "Enter your name: " << endl;
                getline(cin, name);
                cout << "Enter your address: " << endl;
                getline(cin, address);
                cout << "Enter your ID: " << endl;
                cin >> staff_id;
        }
```

```
};
class Doctor :virtual public HospitalStaff {
protected:
	string special;
	int md_lic;
public:
	Doctor()
	{
		special = " ";
		md_lic = 0;
	}
	Doctor(string spec, int mdlc)
	{
		special = spec;
		md_lic = mdlc;
	}
	void inputd()
	{
		cin.ignore();
		cout << "Enter Specilization: " << endl;
		getline(cin, special);
		cout << "Enter Medical License: " << endl;
		cin >> md_lic;
	}
};
class Administrator :virtual public HospitalStaff {
protected:
	string dep;
	string manage;
public:

	Administrator()
	{
		dep = " ";
		manage = " ";
	}
	Administrator(string department, string mng)
	{
```

```
                dep = department;
                manage = mng;
        }
        void input3()
        {
                cin.ignore();
                cout << "Enter the Department: " << endl;
                getline(cin, dep);
                cout << "Enter the Management level: " << endl;
                getline(cin, manage);

        }
};
class Surgeon :public Administrator, public Doctor {
public:
        void display()
        {
                cout << "Name: " << name << endl;
                cout << "Address: " << address << endl;
                cout << "Staff ID: " << staff_id << endl;
                cout << "Specilization: " << special << endl;
                cout << "Medical License: " << md_lic << endl;
                cout << "Department: " << dep << endl;
                cout << "Management Level: " << manage << endl;

        }

};
int main()
{
        Surgeon s1;
        s1.setinput();
        s1.inputd();
        s1.input3();
        cout << "Surgeon Details" << endl;
        s1.display();
        return 0;
}
```

Screenshot:



UMar FArooq-088-lab10-2B

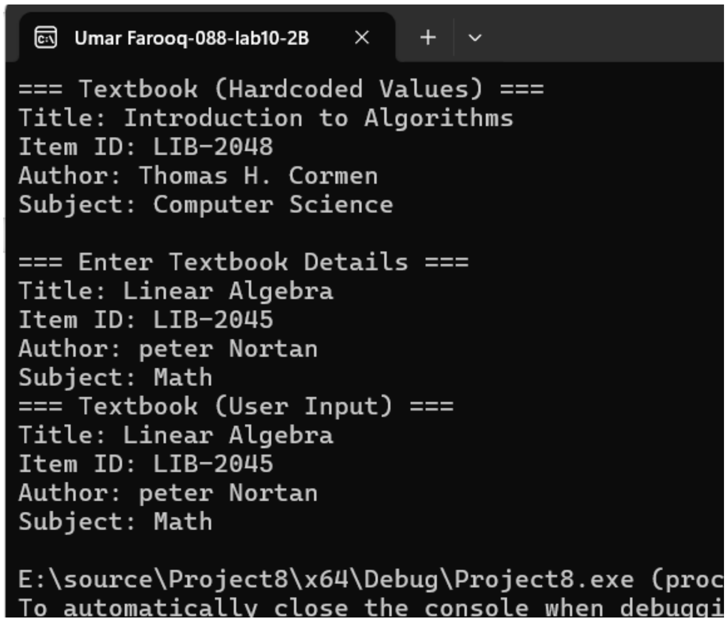
```
Enter your name: Umar FArooq
Enter your address: st#243
Enter your ID: 88
Enter Specilization: Neuro
Enter Medical License: 2335633
Enter the Department: Neuro
Enter the Management level: HOD
Surgeon Details
Name: Umar FArooq
Address: st#243
Staff ID: 88
Specilization: Neuro
Medical License: 2335633
Department: Neuro
Management Level: HOD

E:\source\Project7\x64\Debug\Project7.exe (p
```

Task 02: Implement a library catalog system using multilevel inheritance. Create a base class LibraryItem with attributes title and item_id, and a method display_info(). Derive a class Book from LibraryItem, adding author and display_author(). Further derive a class Textbook from Book, adding subject and display_subject(). The program should create a textbook object and display all levels of information sequentially without any method overriding.

Code:

```
#include <iostream>
#include <string>
using namespace std;
class LibraryItem {
public:
    string title;
    string item_id;

    void display_info()
    {
        cout << "Title: " << title << endl;
        cout << "Item ID: " << item_id << endl;
    }
};

class Book : public LibraryItem {
public:
    string author;
    void display_author()
    {
        cout << "Author: " << author << endl;
    }
};

class Textbook : public Book {
public:
    string subject;

    void display_subject()
    {
        cout << "Subject: " << subject << endl;
    }
};

int main() {
    Textbook tb1;

    cout << "=== Textbook (Hardcoded Values) ===" << endl;
    tb1.title = "Introduction to Algorithms";
    tb1.item_id = "LIB-2048";
    tb1.author = "Thomas H. Cormen";
    tb1.subject = "Computer Science";

    tb1.display_info();
    tb1.display_author();
    tb1.display_subject();
    cout << endl;
    cout << "=== Enter Textbook Details ===" << endl;
    Textbook tb2;

    cout << "Title: ";
    getline(cin, tb2.title);
    cout << "Item ID: ";
    getline(cin, tb2.item_id);
    cout << "Author: ";
    getline(cin, tb2.author);
    cout << "Subject: ";
    getline(cin, tb2.subject);

    cout << "=== Textbook (User Input) ===" << endl;
    tb2.display_info();
    tb2.display_author();
    tb2.display_subject();

    return 0;
}
```

Screenshot:



```
=== Textbook (Hardcoded Values) ===
Title: Introduction to Algorithms
Item ID: LIB-2048
Author: Thomas H. Cormen
Subject: Computer Science

=== Enter Textbook Details ===
Title: Linear Algebra
Item ID: LIB-2045
Author: peter Nortan
Subject: Math
=== Textbook (User Input) ===
Title: Linear Algebra
Item ID: LIB-2045
Author: peter Nortan
Subject: Math

E:\source\Project8\x64\Debug\Project8.exe (proc
To automatically close the console when debuggi
```