Khoi Duong

Prof. Yang

CS360L

7/12/2022

LAB #7

1.

a. True

b. False. Because a is-a relationship is implemented via inheritance

c. False. Because the Car class has has-a relationship with SteeringWheel and Brake class.

d. True

2.

The relative merits of using protected access vs. using private access in base classes:

- Using protected access data helps the derived class manipulate the protected members when it cannot access the base class function.
- Private data members are usually hidden from the base class and can be accessed only through the protected process of the base class.
- The base class is used to access the data member function.
- When performing, it will contain extra function calls. However, it can modify and access the private data member in an indirect manner.
- It assures that the data in the base class is consistent.

3. 4. 5. (Using the same program)

Source code:

publication.h

```
#ifndef PUBLICATION_H
#define PUBLICATION_H
#include <iostream>
using namespace std;

class Publication {
    public:
    string title;
    float price;
    Publication(): title(""), price(0.0) {};
    ~Publication() {};
    virtual void getdata() {
        cout << "Enter title: ";
        cin >> title;
        cout << "Enter price: ";
        cin >> price;
    }
    virtual void putdata() {
        cout << "Title: " << title << endl;
        cout << "Price: " << price << endl;
    }
};
#endif // PUBLICATION_H
```

sales.h

```
#ifndef SALES_H
#define SALES_H

#include <iostream>
using namespace std;

class Sales {
    public:
    float sale[3];
    Sales(): sale{0.0, 0.0, 0.0} {};
    ~Sales() {};
    virtual void getdata() {
```

```
        for (int i = 0; i < 3; i++) {
            cout << "Enter sale " << i + 1 << ": ";
            cin >> sale[i];
        }
    }
    virtual void putdata() {
        for (int i = 0; i < 3; i++) {
            cout << "Sale " << i + 1 << ": " << sale[i] << endl;
        }
    }
};

#endif // SALES_H
```

book.h

```
#ifndef BOOK_H
#define BOOK_H
#include <iostream>
using namespace std;
#include "publication.h"
#include "sales.h"

class Book : public Publication, public Sales {
    public:
    int page_count;
    Book(): Publication(), Sales(), page_count(0) {};
    ~Book() {};
    void getdata() {
        Publication::getdata();
        Sales::getdata();
        cout << "Enter page count: ";
        cin >> page_count;
    }
    void putdata() {
        Publication::putdata();
        Sales::putdata();
        cout << "Page count: " << page_count << endl;
    }
};

#endif // BOOK_H
```

tape.h

```
#ifndef TAPE_H
#define TAPE_H
#include <iostream>
using namespace std;
#include "publication.h"
#include "sales.h"

class Tape : public Publication, public Sales {
    public:
    float play_time;
    Tape(): Publication(), Sales(), play_time(0) {};
    ~Tape() {};
    void getdata() {
        Publication::getdata();
        Sales::getdata();

        cout << "Enter play time: ";
        cin >> play_time;
    }
    void putdata() {
        Publication::putdata();
        Sales::putdata();
        cout << "Play time: " << play_time << endl;
    }
};

#endif // TAPE_H
```

disk.h

```
#ifndef DISK_H
#define DISK_H
#include <iostream>
using namespace std;
#include "publication.h"
#include "sales.h"

class Disk : public Publication, public Sales {
    public:
    enum type {
        CD = 'c',
        DVD = 'd'};
```

```
    char userInput;
    string userOutput;
    Disk(): Publication(), Sales() {}; // default constructor
    ~Disk() {};
    void getdata() {
        Publication::getdata();
        Sales::getdata();
        cout << "CD (c) or DVD (d)? ";
        cin >> userInput;
        switch (userInput) {
            case 'c':
                userOutput = "CD";
                break;
            case 'd':
                userOutput = "DVD";
                break;
        }
    }
    void putdata() {
        Publication::putdata();
        Sales::putdata();
        cout << "CD (c) or DVD (d)? " << userOutput << endl;
    }
};

#endif // DISK_H
```

main.cpp

```
#include <iostream>
using namespace std;
#include "publication.h"
#include "sales.h"
#include "book.h"
#include "tape.h"
#include "disk.h"

int main() {
    Book book1;
    cout << "Input info for book1: " << endl;
    book1.getdata();
    cout << endl;
    cout << "Output info for book1: " << endl;
```

```
    book1.putdata();
    cout << endl;
    Tape tape1;
    cout << "Input info for tape1: " << endl;
    tape1.getdata();
    cout << endl;
    cout << "Output info for tape1: " << endl;
    tape1.putdata();
    cout << endl;
    Disk disk1;
    cout << "Input info for disk1: " << endl;
    disk1.getdata();
    cout << endl;
    cout << "Output info for disk1: " << endl;
    disk1.putdata();
    return 0;
}
```

Run program & result: (next page)

```
PS D:\VS CODE\C C++\CS360L\Lab7> cd
Input info for book1:
Enter title: Divergent
Enter price: 32
Enter sale 1: 1500
Enter sale 2: 1750
Enter sale 3: 2000
Enter page count: 428

Output info for book1:
Title: Divergent
Price: 32
Sale 1: 1500
Sale 2: 1750
Sale 3: 2000
Page count: 428

Input info for tape1:
Enter title: Parasite
Enter price: 15
Enter sale 1: 2500
Enter sale 2: 2950
Enter sale 3: 2300
Enter play time: 325

Output info for tape1:
Title: Parasite
Price: 15
Sale 1: 2500
Sale 2: 2950
Sale 3: 2300
Play time: 325

Input info for disk1:
Enter title: Anabelle
Enter price: 20
Enter sale 1: 1600
Enter sale 2: 1550
Enter sale 3: 1850
CD (c) or DVD (d)? d

Output info for disk1:
Title: Anabelle
Price: 20
Sale 1: 1600
Sale 2: 1550
Sale 3: 1850
CD (c) or DVD (d)? DVD
PS D:\VS CODE\C C++\CS360L\Lab7>
```