

```
#include <list>
#include <iostream>
using namespace std;
#define endl "\n"
```

```
/*
```

a) Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float).

Each of these three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display its data. Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata(), and then displaying the data with putdata().

b) Start with the publication, book, and tape classes of Exercise (a).

Add a base class sales that holds an array of three floats so that it can record the dollar sales of a particular publication for the last three months.

Include a getdata() function to get three sales amounts from the user, and a putdata() function to display the sales figures.

Alter the book and tape classes so they are derived from both publication and sales.

An object of class book or tape should input and output sales data along with its other data.

Write a main() function to create a book object and a tape object and exercise their input/output capabilities.

```
*/
```

```
// Models
```

```
struct Triple{
    float first, second, third;
};
```

```
// Requirement (B)
```

```
class Sales{
private:
    Triple _dollarSales;
public:
```

```
    // Constructor
```

```
    Sales(Triple sales) : _dollarSales(sales){
```

```
}
```

```
// Setters
```

```
void getData(){
    cout << " Enter sales over 3 months:\n";
    cout << " First: "; cin >> _dollarSales.first;
    cout << " Second: "; cin >> _dollarSales.second;
    cout << " Third: "; cin >> _dollarSales.third;
```

```

}

// Getters
void putData()const{
    printf(" Sales(1stMonth = %.2f, 2ndMonth = %.2f, 3rdMonth = %.2f)\n", _dollarSales.first,
_dollarSales.second, _dollarSales.third);
}
};

```

```

// Requirement (A)
class Publication{
private:
    string _title;
    float _price;

public:
    // Constructor
    Publication():_title(""), _price(0.0){}
    Publication(string title, float price):_title(title), _price(price){}

```

```

// Setters
void getData(){
    cout << " Enter Publication info:" << endl;
    cout << " Enter title: "; cin >> _title;
    cout << " Enter price: "; cin >> _price;
}

// Getters
string title(){
    return _title;
}
float price(){
    return _price;
}

void putData()const{
    printf(" Publication(title = %s, price = $%.2f)\n", _title.c_str(), _price);
}
};

```

```

class Book:public Publication, public Sales{
private:
    int _pages;
public:
    // Constructor
    Book():Publication("", 0), _pages(0), Sales({0, 0, 0}){}
    Book(string title, float price, int pages, Triple sales):Publication(title, price), _pages(pages), Sales(sales){}

```

```
// Setters
```

```
void getData(){
    cout << "(Book)(getData)" << "{" << endl;
    Publication::getData();
    cout << "    Enter pages count: "; cin >> _pages;
    Sales::getData();
    cout << "}" << endl;
}
```

```
// Getters
```

```
void putData()const{
    cout << "(Book)(putData)" << endl;
    Publication::putData();
    Sales::putData();
    printf("    Book(pages = %i)\n", _pages);
    cout << "}" << endl;
}
};
```

```
class Tape:Publication, public Sales{
```

```
private:
```

```
    float _playingTime;
```

```
public:
```

```
    // Constructor
```

```
    Tape():Publication("", 0.0), _playingTime(0.0), Sales({0, 0, 0}){}
    Tape(string title, float price, float playingTime, Triple sales):Publication(title, price), _playingTime(
```

```
playingTime), Sales(sales){}
```

```
// Setters
```

```
void getData(){
    cout << "(Tape)(getData)" << endl;
    Publication::getData();
    cout << "    Enter playing time: "; cin >> _playingTime;
    Sales::getData();
    cout << "}" << endl;
}
```

```
// Getters
```

```
void putData()const{
    cout << "(Tape)(putData)" << endl;
    Publication::putData();
    Sales::putData();
    printf("    Tape(playingTime = %.2f minutes)\n", _playingTime);
    cout << "}" << endl;
}
};
```

```
int main(){
```

```
// fIRST Requirement (a)
```

```
Tape Tape1;
```

```
Book Book1;
```

```
Publication Pub1;
```

```
Book1.getData();
```

```
cout << endl;
```

```
Book1.putData();
```

```
cout << endl;
```

```
Tape1.getData();
```

```
cout << endl;
```

```
Tape1.putData();
```

```
cout << endl;
```

```
return 0;
```

```
}
```

```
/* OUTPUT - (A, B)
```

```
PS \GU\OOP\LAB\Assignments\#8\code> c++ main.cpp -o main.exe; .\main.exe
```

```
(Book)(getData){
```

```
    Enter Publication info:
```

```
        Enter title: Book1
```

```
        Enter price: 189
```

```
        Enter pages count: 1800
```

```
    Enter sales over 3 months:
```

```
        First: 10
```

```
        Second: 80
```

```
        Third: 190
```

```
}
```

```
(Book)(putData){
```

```
    Publication(title = Book1, price = $189.00)
```

```
    Sales(1stMonth = 10.00, 2ndMonth = 80.00, 3rdMonth = 190.00)
```

```
    Book(pages = 1800)
```

```
}
```

```
(Tape)(getData){
```

```
    Enter Publication info:
```

```
        Enter title: Tape1
```

```
        Enter price: 80
```

```
        Enter playing time: 1
```

```
    Enter sales over 3 months:
```

```
        First: 90
```

```
        Second: 289
```

```
        Third: 500
```

```
}
```

```
(Tape)(putData){
```

```
    Publication(title = Tape1, price = $80.00)
```

```
    Sales(1stMonth = 90.00, 2ndMonth = 289.00, 3rdMonth = 500.00)
```

Tape(playingTime = 1.00 minutes)

}

*/