

# Inheritance

---

## Synopsis

The learning objectives of this seminar are same as that of the lecture for this week. Please, if you have not done yet, watch the lecture before you attempt this seminar (or do both in parallel).

This laboratory session is intended to enable you understand inheritance and related concepts.

Note: Answers are provided at the end of this file.

## Exercises

### 1

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()**.

### 2

Start with the **publication**, **book**, and **tape** classes of Exercise 1. 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.

## Answers

### 1

```
#include <iostream>
#include <string>
using namespace std;
class publication // base class
{
    private:
        string title;
        float price;
    public:
        void getdata()
        {
            cout << "\nEnter title: "; cin >> title;
            cout << "Enter price: "; cin >> price;
        }
        void putdata() const
        {
            cout << "\nTitle: " << title;
            cout << "\nPrice: " << price;
        }
};
```

# Inheritance

---

class book : **private** publication //Note the 'private' access specifier here. For more, see 'Access Specification' in inheritance at the end

```
{
    private:
    int pages;

    public:
    void getdata()
    {
        publication::getdata();
        cout << "Enter number of pages: "; cin >> pages;
    }
    void putdata() const
    {
        publication::putdata();
        cout << "\nPages: " << pages;
    }
};
```

class tape : private publication // derived class

```
{
    private:
    float time;
    public:
    void getdata()
    {
        publication::getdata();
        cout << "Enter playing time: "; cin >> time;
    }
    void putdata() const
    {
        publication::putdata();
        cout << "\nPlaying time: " << time;
    }
};
```

int main()

```
{
    book book1; // define publications
    tape tape1;
    book1.getdata(); // get data for them
    tape1.getdata();
    book1.putdata(); // display their data
    tape1.putdata();
    cout << endl;
    return 0;
}
```

## 2

```
#include <iostream>
#include <string>
using namespace std;
////////////////////////////////////
class publication
{
    private:
    string title;
    float price;
    public:
    void getdata()
    {
        cout << "\nEnter title: "; cin >> title;
        cout << " Enter price: "; cin >> price;
    }
    void putdata() const
    {
        cout << "\nTitle: " << title;
        cout << "\n Price: " << price;
    }
}
```

# Inheritance

---

```
};

class sales
{
    private:
        enum { MONTHS = 3 };
        float salesArr[MONTHS];
    public:
        void getdata();
        void putdata() const;
};

void sales::getdata()
{
    cout << " Enter sales for 3 months\n";
    for(int j=0; j<MONTHS; j++)
    {
        cout << " Month " << j+1 << ": ";
        cin >> salesArr[j];
    }
}

void sales::putdata() const
{
    for(int j=0; j<MONTHS; j++)
    {
        cout << "\n Sales for month " << j+1 << ": ";
        cout << salesArr[j];
    }
}

class book : private publication, private sales
{
    private:
        int pages;
    public:
        void getdata()
        {
            publication::getdata();
            cout << " Enter number of pages: "; cin >> pages;
            sales::getdata();
        }
        void putdata() const
        {
            publication::putdata();
            cout << "\n Pages: " << pages;
            sales::putdata();
        }
};

class tape : private publication, private sales
{
    private:
        float time;
    public:
        void getdata()
        {
            publication::getdata();
            cout << " Enter playing time: "; cin >> time;
            sales::getdata();
        }
        void putdata() const
        {
            publication::putdata();
            cout << "\n Playing time: " << time;
            sales::putdata();
        }
};
```

# Inheritance

---

```
int main()
{
    book book1; // define publications
    tape tape1;
    book1.getdata(); // get data for publications
    tape1.getdata();
    book1.putdata(); // display data for publications
    tape1.putdata();
    cout << endl;
    return 0;
}
```

## Access Specification in Inheritance

Please see the following example to understand the difference between **public**, **protected** and **private** inheritance in C++.

```
class A {
    public:
        int x;
    protected:
        int y;
    private:
        int z;
};
```

```
class B : public A
{
    // x is public
    // y is protected
    // z is not accessible from B
};
```

```
class C : protected A
{
    // x is protected
    // y is protected
    // z is not accessible from C
};
```

```
class D : private A // 'private' is default for classes
{
    // x is private
    // y is private
    // z is not accessible from D
};
```

Click this to play with the above code: <https://onlinegdb.com/wvCWeHkOr>