

### Assignment - 3

Name: Tanmay Karmarkar  
Roll No. 21143

Performance Date:  
Submission Date:

Demonstrate reusability of code through inheritance.

#### Problem Statement :

Imagine a publishing company which does marketing for the book and audio cassettes. Create a class publication that stores title & price of a publication. From this, derive 2 classes book, which stores page count & tape which stores play time, in minutes.

Write a program that instantiates the book & tape classes, allows user to enter data & display data members.

#### Theory:

##### Inheritance:

C++ supports the concept of inheritance (reusability) once a class has been written & tested, it can be adapted by other programmers to suit their requirements. This is done by creating classes, reusing properties of existing ones. The mechanism of deriving a new class from an old one is called inheritance.

##### Base & derived classes:

When creating a class, instead of writing completely new data members & member functions, the programmer can designate that the new class should inherit members of existing class.

This existing class is called base class & the new class is called derived class. A class can be derived from more than one class.

## Ex - Example

### Access control:

A derived class can access all the non-private members of its base class. Thus base class members that should not be accessible to derived class should be declared in private.

| Access       | Public | Private | Protected |
|--------------|--------|---------|-----------|
| Same derived | y      | y       | y         |
| outside      | y      | n       | n         |

### Algorithm:

1. Start
2. Define class Publication with data members title & price in public & also pricebook & titlebook arrays.
3. Define class Book publically inherited from publication.
4. Declare pgmt & pagecount array in public section.
5. Define functions in both derived class for taking input, update, delete, search & display.
6. Add EH so when wrong input is given all parameters are set to 0.
7. Follow the same steps as class Book for class type.
8. Stop.

| Input   | Exp. O/P      | Actual O/P       | Status |
|---|---------------|------------------|--------|
| 1. Create Book<br>Title : Abc<br>Price : 100<br>Pages : 50                  | Invalid input | Setting all to 0 | Pass   |
| 2. Create Book<br>Title : Bcd<br>Price : -1                                 | Invalid input | Setting all to 0 | Pass   |
| 3. Create Book<br>Title : xyz<br>Price : 200<br>Pages : -3                  | invalid input | Setting all to 0 | Pass   |
| 4. Search Book<br>Title : abc   | Not found     | Not found        | Pass   |
| 5. Update Book<br>Title : abc   | Not found     | Not found        | Pass   |
| 6. Update Book<br>Title : Abc<br>New : abc<br>Price : 50<br>Pagecount : 100 | -             | -                | Pass   |
| 7. Create Tape<br>Title : Abc<br>Price : 100<br>Pages Time : 50             | -             | -                | Pass   |
| 8. Create Tape<br>Title : abc<br>Price : -1                                 | Invalid input | Setting all to 0 | Pass   |

|     |                           |                         |                         |      |
|-----|---------------------------|-------------------------|-------------------------|------|
| 9.  | Delete Book<br>title: abc | Deleted<br>Successfully | Deleted<br>Successfully | Pass |
| 10. | Delete BOOK<br>title: abc | Not found               | Not found               | Pass |

most likely to be  
tuguri big - white

and up until I  
got there

1999-09-10 1999-09-10 1999-09-10 1999-09-10