

OO mini-project

Problem Statement:

The task is to create a C++ OO program modelling a simple library system. The library in question stores different types of media (publications): books, journals, music CDs, film DVDs.

- A book includes the following details: *author, title, ISBN, publisher, refNo*.
- A journal contains: *title, date, editor, refNo*.
- A music CD contains: *title, category, artist, number of tracks, list of tracks, refNo*. A track contains: *title, duration*.
- A DVD contains: *title, category, director, number of actors, list of actors, duration, refNo*.

In addition to storing the publications, the library has a name and can carry out the following services:

- 1) A publication can be borrowed (if it is not already on loan)
- 2) A publication on loan can be returned
- 3) A new publication can be added to the library collection
- 4) Display one of: all books, journals, CDs, DVDs in the library
- 5) Display the details of a particular book, journal, CD or DVD (for CDs it should also display the total duration of the CD)
- 6) Display all the publications in the library
- 7) Display all the publications that are on loan
- 8) Display how many of each type of publication in the library (including those on-loan)

Design and implement a system to model the library system and write a main program to test your implementation. It should present a menu of options 1-8, allow the user to provide the relevant input data and then print out the resulting confirmation message/information to the screen. For simplicity you can assume that there is only one copy of each publication in the library.

Possible approach: design first (on paper write down the classes involved - what data and functionality should they provide, what are the relationships between the classes), then implement incrementally testing the functions as you go, each 'release' is a working solution but with restricted functionality, advance towards the solution with complete functionality.

Complete by Friday 7th October to demo in class.