Assignment 2 - part 1

Deadline	Anytime before Sunday, 29th April 2018, 15:00
Evaluation	10 marks –which is 5% of your final grade
Late Submission	5% per hour (or fraction of hour) it is late
Team	The assignment can be done individually or in pairs (of at most 2 students)
Purpose	Practice with inheritance and polymorphism.

Problem to solve

Assume you have to write software to be used by Massey University library. Assume that the only items that can be borrowed from the library are books, journals and DVDs. We will call them `Item`(s). In this assignment you have to write a simple hierarchy for items that the library lends to public. You will extend the `Item` class given in file `a2p1_classes.h` to three more classes `Book`, `DVD` and `Journal` specific to the properties of these items.

The library also charges fees for overdue or missing items. The fee for missing items is a flat $200 and the fee for overdue items is $5 a day. `Book` items also have a special status indicated by the letter `S` at the beginning of the call number. If any fee is owed, special books add a surcharge of $30 on top of such fee.

Name your module `a2p1_classes.cpp`
Hand-in: Submit `a2p1_classes.cpp` electronically using the submission form on STREAM.

If you have any questions about this assignment, please ask the lecturer before the assignment is due.

Problem statement

You are given the following classes:

1. `Status`: is an `enum` class containing all possible statuses of a library item.
2. `Item`: is an abstract class containing some generic properties of a library item. It contains the following member variables and methods:
   - **`private`** `status`: status of the item
   - **`private`** `title`: title of the item
   - **`private`** `daysOverDue`: number of days overdue
   - **`private`** `callNumber`: call number

- A **public** constructor that takes the following arguments in order: `callNumber`, `title`, `status`, `daysOverDue`.
- Several **public** `getters`
- **public** `print`: a pure virtual function that returns a string in the appropriate format containing all information about the object
- **public** `getFees`: a virtual function that returns the total fee owed according to library's rule.

3. There is also the declaration of function **`printStudentInfo()`** that you need to implement.

Read the documentation in the header file, `a2p1_classes.h`, you are given on Stream, for more details. You cannot make any changes to the file `a2p1_classes.h` and you will not send the file `a2p1_classes.h` for marking, but you will use it when you will check your solution. Also, you will not create your own main function. Instead, your classes will be imported into and used by a main function similar to the following:

```
#include <iostream>
#include "a2p1_classes.cpp"

using namespace std;

/**
 * Total number of items
 */
const int NUM_ITEMS = 12;

int main() {

  printStudentInfo();

  Item * items[NUM_ITEMS] = {
    new Book("S529.030MAR", "The adventures of Tom Sawyer", "Mark Twain", 500,
Status::MISSING, 0),
    new Book("S759.010BEN", "Bullies", "Ben Shapiro", 336, Status::OVERDUE, 2),
    new Book("123.010MIL", "Dangrous", "Milo Yiannopolous", 288, Status::MISSING, 10),
    new Book("S205.682CON", "Sherlock Holmes", "Conan Doyle", 68, Status::ON_LOAN, 3),

    new Journal("123.456SCI", "Scientific American", "XXI", 4, Status::MISSING, 10),
    new Journal("382.150ORN", "Ornothologist", "XIV", 1, Status::AVAILABLE, 0),
    new Journal("842.003NAT", "National Geographic", "XXIX", 2, Status::OVERDUE, 8),
    new Journal("153.227NAT", "Computational Biology", "VII", 4, Status::AVAILABLE, 8),

    new DVD("235.707HAY", "Spirited Away", "Hayao Miyazaki", Status::AVAILABLE, 0),
    new DVD("987.654SHI", "Your Name", "Shinkai Mikoto", Status::OVERDUE, 6),
    new DVD("258.147SHO", "Koe No Katachi", "Shochiku", Status::ON_LOAN, 0),
    new DVD("S963.159TIM", "Big Fish", "Tim Burton", Status::OVERDUE, 7)
  };

  int totalFees = 0;

  for (int i=0; i<NUM_ITEMS; i++) {
    Item * item = items[i];
    cout << item->print() << endl;
    totalFees += item->getFees();
  }
```

```
  cout << "Total fees: " << totalFees << endl;
}
```

Keep in mind this is just an example. The actual `main` function being used to evaluate your module will be different.

You will create the following derived classes:

1. Class `Book` that has the following additional members:
   - **`private`** `author`: Name of the author
   - **`private`** `numPages`: Number of pages
   - A **`public`** constructor that takes the following arguments in order: `callNumber`, `title`, `author`, `numPages`, `status`, `daysOverDue`.
2. Class `Journal` that has the following additional members:
   - **`private`** `volume`: The journal's volume
   - **`private`** `frequency`: How many times this journal is published every year
   - A **`public`** constructor that takes the following arguments in order: `callNumber`, `title`, `volume`, `frequency`, `status`, `daysOverDue`.
3. Class `DVD` that has the following additional members:
   - **`private`** `producer`: Name of the producer
   - A **`public`** constructor that takes the following arguments in order: `callNumber`, `title`, `producer`, `status`, `daysOverDue`.

Sample output

Your new classes must make the given `main` function compile and print out the following output (next page) EXACTLY:

```
**********************************
* 159.234 Assignment 2 Part 1
* Name: Jane Doe, John Smith
* ID: 17012345, 16234567
**********************************

Book #S529.030MAR, title: "The adventures of Tom Sawyer" by Mark Twain, pages: 500, status: Missing, overdue fee: 230
Book #S759.010BEN, title: "Bullies" by Ben Shapiro, pages: 336, status: Overdue, overdue fee: 40
Book #123.010MIL, title: "Dangerous" by Milo Yiannopolous, pages: 288, status: Missing, overdue fee: 200
Book #S205.682CON, title: "Sherlock Holmes" by Conan Doyle, pages: 68, status: On Loan, overdue fee: 0
Journal #123.456SCI, title: "Scientific American" volume XXI, frequency: 4, status: Missing, overdue fee: 200
Journal #382.150ORN, title: "Ornithologist" volume XIV, frequency: 1, status: Available, overdue fee: 0
Journal #842.003NAT, title: "National Geographic" volume XXIX, frequency: 2, status: Overdue, overdue fee: 40
Journal #153.227NAT, title: "Computational Biology" volume VII, frequency: 4, status: Available, overdue fee: 0
DVD #235.707HAY, title: "Spirited Away" produced by Hayao Miyazaki, status: Available, overdue fee: 0
DVD #987.654SHI, title: "Your Name" produced by Shinkai Mikoto, status: Overdue, overdue fee: 30
DVD #258.147SHO, title: "Koe No Katachi" produced by Shochiku, status: On Loan, overdue fee: 0
DVD #S963.159TIM, title: "Big Fish" produced by Tim Burton, status: Overdue, overdue fee: 35
Total fees: 775
```

Your module will be compiled, run and evaluated by a machine. So besides your name(s) and ID(s), the output must be EXACTLY the same, character to character.

Requirements

What you must and must not do

1. Your module MUST be named `a2p1_classes.cpp`. If you use a different name, you will get zero.
2. File `a2p1_classes.cpp` must not contain any extra class, but it can contain helper functions if you want to.
3. Your module MUST NOT require any user interaction.
4. You MUST NOT make any changes to the given header file (`a2p1_classes.h`).
5. You MUST NOT include any non-C++ standard library header files. E.g. window.h is a Windows-specific header and you MUST NOT include it.
6. You MUST NOT use the exit function.
7. You MUST follow the same style and format that are used in the header.
8. Your module SHOULD be organised as followed:
    - ✓ Header (Your name(s), ID (s), short description for the program, etc)
    - ✓ All included files/libraries
    - ✓ Class declaration
    - ✓ Class definitions
    - ✓ All functions and classes MUST be properly documented following Doxygen format. See http://www.yolinux.com/TUTORIALS/LinuxTutorialC++CodingStyle.html for detail. You will also learn how to document your code in tutorials.
9. You MUST NOT use dynamic memory allocation, or raw pointers
10. You MUST NOT use advanced features not covered by the course by the due date

Miscellaneous

1. When working in pair, send one solution file per team.
2. The assignment will be discussed on Wednesday lecture before the assignment is due and solutions will be discussed on Monday lecture after the due time.
3. Marks will be allocated for: correctness, completeness, consistent coding style, sensible structure (simple and clear solution, good use of helper functions), good understanding of class inheritance, good documentation.
4. Using goto, non-constant global variables or C-like I/O constructs (i.e printf fprintf, scanf, FILE*,etc) is not allowed and it will be penalised. Only const global variables are allowed.
5. Programs that do not run or do not compile in the (Albany) labs, using gcc(SciTe), get 0 marks.
6. The program must be your own work. Please be aware that you might be asked to explain to your lecturer how your program works. If you cannot explain it, then it is not yours and you will get 0 marks for that assignment. Attributing someone else's work as your own is plagiarism, and it is a violation of Massey University policy. We might file an official complaint against any student who we believe has committed plagiarism.
7. Suspicious similar solutions will all get 0 marks-see also point 6 above