Object-Oriented Programming
(BE-CS-F20 Morning & Afternoon)
Assignment # 2

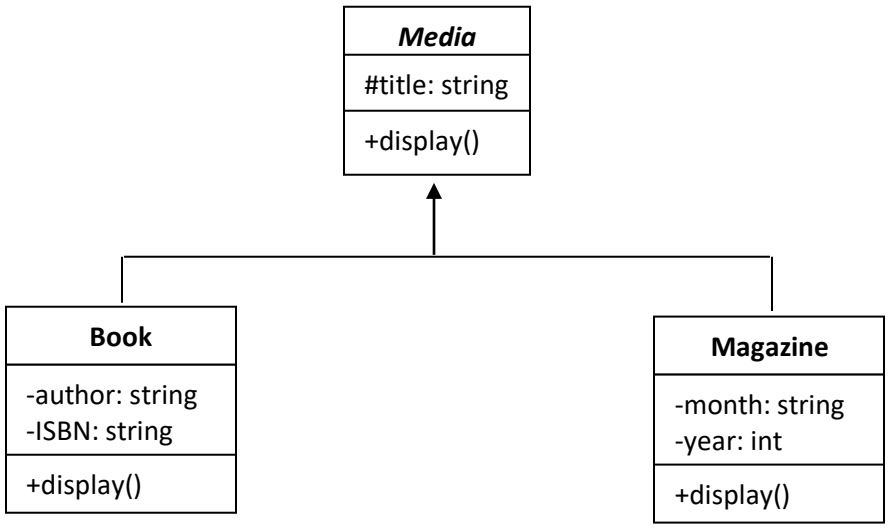
Important Instructions:

- Indent your code properly.
- Use meaningful variable and function names. Follow the naming conventions.
- Make sure that there are no dangling pointers and memory leaks in your program.

Note: *The declaration and implementation of each class that you implement should be in separate .h and .cpp files.*

Task # 1.1

In this task, you are required to implement the following inheritance hierarchy:



Declare and implement the *abstract* class **Media**. This class will have a protected member variable **title** (of string type) to store the title of the media item. Apart from the overloaded constructor, Media class will a *pure virtual function* **display().**

Inherit two classes from Media class, namely: **Book** and **Magazine**.

The **Book** class will have two strings to store the **author name** and **ISBN** of the book. Apart from the overloaded constructor, this class will implement the **display()** function which will display all attributes of a Book on screen in a neat and readable way.

The **Magazine** class will have a string to store the **month name** and an integer to store the **year of publication** of the magazine. Apart from the overloaded constructor, this class will also implement the **display()** function which will display all the attributes of a Magazine on screen in a neat and readable way.

Now, implement a main function which should ask the user how many media items the user wants to create, and store the value entered by user in an integer variable **n**. Then, your program will dynamically allocate an array of **Media*** of size **n**.

After that, your program will ask the user to create **n** Media items. The user should be asked to enter 1 if he/she wants to create a Book and 2 if he/she wants to create a Magazine. Once the choice has been entered, your program should ask for all the necessary attributes for creating that item (Book or Magazine). Then, that item should be dynamically allocated and its address should be stored in the array of **Media***.

Once all Media items have been created, your program should traverse the array of **Media*** to display the details of each item on screen.

At the end, your program should properly deallocate all the dynamically allocated memory.

Task # 1.2

In order to accomplish this task, you will first need to implement a public member function **int getYear()** in the Magazine class.

Modify the program that you wrote in Task # 1.1 and implement a global function:

void searchByYear (Media, int)**

which takes the array of **Media*** and its **size** as parameters. The function should ask the user to enter a year, then it should search the array for all the **magazines** of that year and display their details on screen.

Now, call the above function from the main function to search the array **Media*** once all Media items have been created.

Task # 1.3

Add another class **CD** to the inheritance hierarchy. The **CD** class will have an integer member variable to store its **capacity** in MBs. Apart from the overloaded constructor, this class will also implement the **display()** function which will display all the attributes of a CD on screen in a neat and readable way.

Also modify the main function to allow user to create a CD by entering the option 3.

Task # 1.4

Add a **Shelf** class to store a list/collection of **Media** items. It will have the following declaration:

```
class Shelf
{
private:
        Media** items;
        int maxSize;
        int currSize;
public:
        Shelf (int);
        void insert (Media*);
        void displayContents ();
        ~Shelf();
};
```

The overloaded constructor will take an integer value as argument and initialize the **maxSize** to that value, and initialize **currSize** to 0. Constructor will also dynamically allocate an array of **Media*** through the member variable **items**.

The member function **insert(Media*)** will take a **Media*** as parameter, and store that pointer into the next available index in the **items** array (use the member variable **currSize** to determine the next available index in the array). This function will also increment **currSize** to indicate the updated size of array.

The member function **displayContents()** will display the details of all the items that are currently stored in the items array. This will be accomplished simply by calling the display member function on each Media item stored in the **items** array.

The destructor of **Shelf** class will deallocate the dynamically allocated array which was allocated by the constructor.

Now, implement a main function which should ask the user how many Media items the user wants to create, and declares a **Shelf** object to store those many items. After that the user should be asked to create those many Media items. After the creation of each Media item (Book, Magazine, or CD), it should be inserted into the Shelf. Once all the items have been inserted, their details should be displayed by calling the **displayContents()** function.