Assignment 2-part 1

Deadline:	**Anytime before** Sunday, 8th May 2016, 23:59 (midnight)
Evaluation:	10 marks – which is 5% of your final grade
Late Submission:	5% per hour (or fraction of hour) it is late
Purpose:	Practice with inheritance and polymorphism.

Problem to solve:
You will design a class hierarchy to work with date objects as described below. For submission reasons you must write all your class declarations & definitions in one file that should be named ***a2p1.h***

Requirements:
In what follows assume that YYYY MM DD, are three integers and YYYY stands for a year number, MM for a month number and, DD stands for a day number (e.g.: 2002 9 16 stands for year 2002, month September and day 16; for more info about standard date formats see:
https://en.wikipedia.org/wiki/Date_format_by_country
For this assignment you will consider that a date can be given by

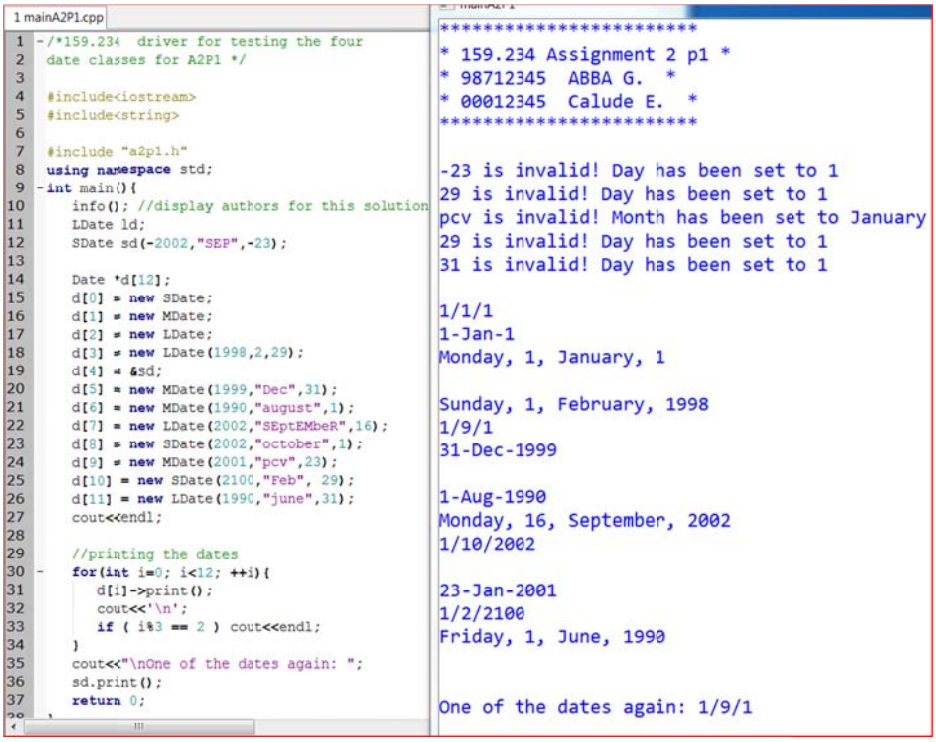
- three integers: YYYY MM DD or by
- an integer YYYY, a string specifying the name of the month and another integer DD; the name of the month may be short - like "Sep." or long -like "September".

You have to implement a hierarchy of date classes with a base class named **`Date`**. The **`Date`** class must have a pure virtual member function **`print`**. From **`Date`** class derive three classes: **`SDate`**, **`MDate`**, and **`LDate`**. Each class will display a date in a different format as explained below.
`SDate`'s `print` must display a date using a short format: DD/MM/YYYY (e.g: 10/9/2016).
`MDate`'s `print` must display a date using a medium format: DD-short month name-YYYY (e.g. 10-Sep-2016). `LDate`'s `print` must display a date using a long format: day name, day number, long month name, year (e.g. Saturday, 10, September, 2016).
Note: The classes you design for this assignment must provide at least: default constructors, custom constructors, and a print member function. See the example in Figure 1 for a possible driver program and the corresponding output produced.

Hand-in:
Submit **a1p2.h** electronically using STREAM.

Miscellaneous:
1. Do not include a main function in your a2p1.h file
2. Do not send the file you used to test your classes.
3. Programs that do not compile in the lab, using gcc, get 0 marks.
4. Marks will be allocated for: correctness, completeness, use of C++ constructs, **good OOP style**-as presented in lectures & labs, good structure for the solution, documentation, and clear on screen output display.
5. Using goto, **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.
6. Write YOUR ID NUMBER(S), and YOUR **FAMILY** NAME(S) first, assignment number, what the program does at the beginning of the file you send electronically and *at least* comment each function.
7. When working in pairs, send one solution file per pair.
8. For algorithms to find the weekday for a give calendar date see:
`https://en.wikipedia.org/wiki/Determination_of_the_day_of_the_week`

If you have any questions about this assignment, please ask the lecturer before its due time!



1 mainA2P1.cpp

```
/*159.234  driver for testing the four
date classes for A2P1 */

#include<iostream>
#include<string>

#include "a2p1.h"
using namespace std;
int main(){
    info(); //display authors for this solution
    LDate ld;
    SDate sd(-2002,"SEP",-23);

    Date *d[12];
    d[0] = new SDate;
    d[1] = new MDate;
    d[2] = new LDate;
    d[3] = new LDate(1998,2,29);
    d[4] = &sd;
    d[5] = new MDate(1999,"Dec",31);
    d[6] = new MDate(1990,"august",1);
    d[7] = new LDate(2002,"SEptEMbeR",16);
    d[8] = new SDate(2002,"october",1);
    d[9] = new MDate(2001,"pcv",23);
    d[10] = new SDate(2100,"Feb", 29);
    d[11] = new LDate(1990,"june",31);
    cout<<endl;

    //printing the dates
    for(int i=0; i<12; ++i){
       d[i]->print();
       cout<<'\n';
       if ( i%3 == 2 ) cout<<endl;
    }
    cout<<"\nOne of the dates again: ";
    sd.print();
    return 0;
}
```

```
************************
* 159.234 Assignment 2 p1 *
* 98712345  ABBA G.  *
* 00012345  Calude E.  *
************************

-23 is invalid! Day has been set to 1
29 is invalid! Day has been set to 1
pcv is invalid! Month has been set to January
29 is invalid! Day has been set to 1
31 is invalid! Day has been set to 1

1/1/1
1-Jan-1
Monday, 1, January, 1

Sunday, 1, February, 1998
1/9/1
31-Dec-1999

1-Aug-1990
Monday, 16, September, 2002
1/10/2002

23-Jan-2001
1/2/2100
Friday, 1, June, 1990


One of the dates again: 1/9/1
```

Figure 1, A possible main function and the corresponding output.