

Fall Semester 2019

# Library Publication Storage and Retrieval Application

V0.5 (Milestone 1) - draft

When Books and other publications arrive in a library, they should be tagged and put on shelves, so they are easily retrievable to be lent out to those who need it.

Your task is to design an application that receives the publications and stores them into the system with the information needed for their retrieval.

Later, each publication can be lent out to members of the library with a due date for return.

Before we start developing the application, we need to have few classes developed to help us with the dates in the system and also the user interface of the application.

## OVERVIEW OF THE CLASSES TO BE DEVELOPED FOR MILESTONE 1

These classes are:

### **Date**

A class that encapsulates year, and month and day values for Date stamps, comparison and Date-IO purposes.

### **MenuItem**

A class that hold a text Item; (an option or title to be displayed) in a menu to be selected by the user. This is a fully private class that is only accessible by Menu (see next class)

### **Menu**

A class that has several **MenuItems** to be displayed so the user can select one of them for an action to be executed in the program. This class is a friend of **MenuItem** class.

## THE DATE CLASS

The Date class was partially implemented by another programmer that left the company and your responsibly is to reuse the parts she developed to complete the implementation:

The date class encapsulates the following values:

- Year; an integer between the year 1500 till today
- Month, an integer between 1 and 12
- Day, an integer between 1 and the number of days in the month.
- Error code; an integer which holds the code that corresponds to an error that recently happened or ZERO if the date object is valid and ready to be used.
- Current year; an integer that is automatically initialized to the current date of the system for validation purposes when a Date object is instantiated.

The Date module in files Date.h and Date.cpp is well documented and is placed in the project directory.

Study it and learn what each constant, variable and member function do and then using those functions and your knowledge of istream; cin and cout add the following member functions to the Date class:

```
std::istream& read(std::istream& is = std::cin);
```

This function reads a date from console in following format YYYY/MM/DD as follows:

- Clear the error code by setting it NO\_ERROR
- Read the year, the month and the day member variables using istream and ignore a single character after the year and the month values to bypass the Slashes  
*Note that the separators do not have to be Slash characters "/" but any separator that is not an integer number.*
- Check and see if istream has failed. If it did fail, set the error code to CIN\_FAILED and clear the istream.  
If not, validate the values entered.
- Flush the keyboard
- Return the istream object

```
std::ostream& write(std::ostream& os = std::cout) const;
```

If the Date object is in a "bad" state or (it is invalid) print the "**dateStatus()**".

otherwise the function should write the date in the following format using the ostream object:

- Prints the year
- Prints a Slash "/"
- Prints the month in two spaces, padding the left digit with zero, if the month is a single digit number
- Prints a Slash "/"
- Prints the day in two spaces, padding the left digit with zero, if the day is a single digit number
- Makes sure the padding is set back to spaces from zero

- Returns the ostream object.

## Operator overloads: (do not use friend)

Overload the following comparison operators to compare two dates.

Use the return value of the `daysSince0001_1_1` member function to compare two dates:

```
bool operator==
bool operator!=
bool operator>=
bool operator<=
bool operator<
bool operator>
```

Use the return value of the `daysSince0001_1_1` member function to overload the `int operator-`

to return the difference in number of days between two dates.

Example:

```
Date
    D1(2019, 12, 02),
    D2(2019, 11, 11);
int days = D1 - D2;
```

“days: in the above code snippet will hold the value 21.

## Bool cast overload:

Overload the Boolean cast so if a date is casted to `bool`, it will return true if the date is valid and false if it is not.

## Helper operator overloads:

Overload the following helper operator overloads to have the Date class compatible with cin and cout, input and output operations:

```
operator<< (for cout)
operator>> (for cin)
```

## DATE FUNCTIONS THAT ARE ALREADY IMPLEMENTED:

### Private functions:

```
int daysSince0001_1_1()const; // returns number of days passed since the date 0001/1/1
bool validate();             /* validates the date setting the error code and then
                             returning the result that is true, if valid, and
                             false if invalid. */
void errCode(int theErrorCode); // sets the error code
int curYear()const;           // returns the current system year
bool bad()const;              // return true if
int mdays()const;           // returns the number of days in current month
void setToToday();            // sets the date to the current date (system date)
```

### Public Functions and constructors:

```
Date();                      // creates a date with current date
Date(int year, int mon, int day); /* creates a date with assigned values
```

```

                                then validates the date and sets the
                                error code accordingly */
int errCode()const;           // returns the error code or zero if date is valid
const char* dateStatus()const; /* returns a string corresponding to the current status
                                of the date */
int currentYear()const;       // returns the m_CUR_YEAR value;

```

## DATE TESTER PROGRAM AND EXECUTION SAMPLE

Write your own tester or use **dateTester.cpp** to make sure your Date Module works correctly.

Compile your **Date** module with either **dateTester.cpp** or **dateSubmissionTester.cpp** for pre-submission test.

You should complete the coding for the Date module in two days.

For execution sample run any of the following commands on matrix:

```
~fardad.soleimanloo/244/ms1_date/dateTester
```

```
~fardad.soleimanloo/244/ms1_date/submissionTester
```

## THE MENUITEM AND THE MENU CLASS

To be continued.

## DUE DATE FOR MILESTONE 1

Monday November 18<sup>th</sup>, 2019

## SUBMISSION INSTRUCTIONS

TBA