Assignment 5 – User-Defined Classes/Pointers

Deadline: **Sunday Nov. 25 at 11:55PM**
Type: **Group Assignment**
Weight: **This assignment is worth 5% of your final grade**

Notes:

- Please do not submit exe files
- All submissions must be done through Moodle

Marking Scheme:

- Program correctness (80%)
- Program clarity (output format, comments, completeness, readability) (20%)

Exercises:

Q1. (25 marks) Consider the following class:

```
/* date.h */
#ifndef DATE_H_
#define DATE_H_

class Date {
  public:
      Date(int=1, int=1, int=2000); // sets day, month, year
      void setDate(int, int, int);  // sets the date
      void printDate() const;  // prints date to the screen

  private:
      int day;
      int month;
      int year;
};

#endif /* DATE_H_ */

/* date.cpp */
#include <iostream>
#include "date.h"

using namespace std;


// Constructor
Date::Date (int d, int m, int y)
{
    day = d;
    month = m;
    year = y ;
}
```

```
// sets date
void Date::setDate(int d, int m, int y)
{
    day = d;
    month = m;
    year = y ;
}

// prints date
void Date::printDate() const
{
 cout << month << "/" << day << "/" << year << "\n";
}
```

a. Add a new function to the class Date that returns the month in letters (e.g., January, February, etc.).

b. Add a function that displays “Happy New Year!” when the data is January 1.

c. Add a function that compares two dates.

d. Test the new functions of the class Date by creating **dynamically (i.e., using pointers)** two objects of the class Date and invoking the new member functions on both objects.

Deliverables: date.h, date.cpp, testdate.cpp.

Q2. (25 marks) Modify Q1 of Assignment 4 by changing the dates and times of the corresponding data members to objects of the classes Date (from Q1 above) and Time (from the lectures). Modify the member functions accordingly. Test your class by **dynamically** creating two Flight objects and calling on them the class member functions.

Deliverables: date.h, date.cpp, time.h, time.cpp, flight.h, flight.cpp, testflight.cpp.

Q3. (25 marks) Write a program that takes a matrix of size N x N as input and sorts its elements as shown in the following example. Do not use C++ standard library for sorting. Use pointers and pointer arithmetic to represent the matrix and to navigate through it.

65	15	82	105
70	78	96	90
45	1	260	240
256	200	265	300

⇨

1	15	45	65
70	78	82	90
96	105	200	240
256	260	265	300

Deliverable: q3.cpp (and header files if you decide to use functions or classes)