

# MCIS-4130

## Homework Assignment 6

### **Purpose:**

This exercise provides opportunities to explore building a class with instances of other classes inside it.

### **What to hand in:**

Hand in your test program for the Birthday class; include your header file, your implementation files, and test run output of the program. Include an answer for the question!

### **Problems:**

#### **A. Birthday Class**

In this exercise you are to sketch the basis of a class containing instances of other classes as its members. You will also provide a test program for it. When creating instances of Birthday, what order are the classes constructed in? The whole class interface is provided as follows:

```
class Birthday
{
public:
    Birthday(const DUString& s, const Date& d);
    Birthday (const char *st, int day, int month, int year);
    Birthday(const Birthday&);

    ~Birthday();
    Birthday& operator = (const Birthday&);

    int Equal ( const Birthday &) const;
    int Equal ( const Date& ) const;
    int Equal ( const DUString& ) const;
    void Print (ostream& os) const;

private:
    DUString name;
    Date birthdate;
};
```

### **Notes:**

For part A, ensure your constructors behave appropriately. Note that the Equal operators are just what are needed for problem 7, and that they could easily overload operator ==. Be sure to write a test program to check it! You should use your date class from HW5, and a DUString class like that discussed in class (you are free to use the C++ string class as well). You will want to use the Date and DUString (or other string class, if you've chosen a different one) functions for checking equality and printing to make this one quite easy! You may use a small subset of the problem 7 data to test this with, if you wish.

### **Evaluation criteria:**

For user interface (if any), be sure that premature EOF or invalid input does not cause the program to perform improperly or go into an infinite loop.

For class definitions, be sure to separate the definition into header and body parts. Please do not provide inline function definitions in the header. Ensure that class functions have the proper visibility (*public* or *private*), and that proper constructors, destructors, and other special functions are provided where necessary.

15	Proper Birthday class header
45	Proper Birthday class implementation
45	Proper test program and output, and answer to the question in the Problem section <b><i>Note: If your program doesn't compile, hand in the error listing as the results!</i></b>