CST 238 Spring 2020, Dr. Gross, CSUMB

Homework 5: Dynamic Arrays and Class Redux

Due: Tuesday, Mar 17, 11:55pm

How to turn in:

- Write two programs and submit them on iLearn
- If you use repl.it, the files will be named main.cpp; you will need to rename them
- The names are in the assignment; incorrect names will be graded as 0

Part 1. Array Merge (10 pts)

Write a C++ program that conducts the following operations:

1. Read in the first number `n` from a file and allocate a dynamic array that can hold `n` int values
2. Read the next `n` int values from the file into that dynamic array (note that the values must be entered in ascending order)
3. Read in the next number `m` from a file and allocate a dynamic array that can hold `m` int values
4. Read the next `m` int values from the file into that dynamic array (note that the values must be entered in ascending order)
5. Create a new dynamic array of size `n + m` and merge the contents of the first and second arrays *in order*
6. Print the array values pointed by one-by-one
7. Deallocate the storage of all arrays

For the following example file:

```
5
5  5  10  20  30
4
15  25  40  50
```

The following shows a sample run of your program:

```
Combined array: 5  5  10  15  20  25  30  40  50
```

Final Notes:

- Read the sample run carefully to understand the requirements of the problem correctly
- This program should be uploaded to iLearn as hw05-1.cpp

Part 2. 24 Hour Clock (15 points)

Write a C++ program that creates a class called Time to hold hours and minutes using military time (https://en.wikipedia.org/wiki/24-hour_clock). The class should:

- Allow the programmer to create a default Time object (hour and minute set to 0)
- Allow the programmer to create a Time object setting the hours and minutes in the constructor
- Automatically set the hours and minutes of any invalid time to 0
- Allow the programmer to get a string of the Time object using the standard format (e.g., `1:23AM`)

- Allow the programmer to get a string of the Time object using the military format (e.g., `0123`)
- Allow two Time objects to be added together using a plus operator, returning a new time
- Times that roll over to a new day should be printed accurately (e.g, `2345` + `0016` should be `0001`)

Hint 1: To build strings with integers, you will need the `sstream` library and the `sstream` class.

Hint 2: To format properly, you will need the `iomanip` library and the `setw()` and `setfill()` functions. This is only worth 2 points, so do this absolutely last.

Here is a sample main() function:

```
int main() {
    Time t1(12, 55);
    Time t2(29, 34);
    Time t3(3, 45);
    Time t4(22, 33);
    cout << t1.getStandard() << " - " << t1 << endl;
    cout << t2.getStandard() << " - " << t2 << endl;
    cout << t3.getStandard() << " - " << t3 << endl;
    cout << t4.getStandard() << " - " << t4 << endl;
    Time t5 = t3 + t4;
    cout << t5.getStandard() << " - " << t5 << endl;
}
```

This is the sample output for the sample program:

```
12:55PM - 1255
12:00AM - 0000
 3:45AM - 0345
10:33PM - 2233
 2:18AM - 0218
```

The class declaration should be:

```
class Time {
public:
    Time();
    Time(int hours, int minutes);
    string getStandard() const; // return time as 12:34AM
    string getMilitary() const; // return time as 0034
    int getHours() const { return myHours; };
    int getMinutes() const { return myMinutes; };
    const Time operator + (const Time & time2) const;
    friend ostream & operator<<(ostream & out, const Time & t); //<< getMilitary()
private:
    int myHours;
    int myMinutes;
};
```

Final Notes:

- Read the sample run carefully to understand the requirements of the problem correctly
- This program should be uploaded to iLearn as hw05-2.cpp