**CS 172 Exam 1 (Spring, 2016)**

**General instructions:**

You are allowed *to look at* any of your previous programming solutions for this exam. You may also look at <http://www.cplusplus.com/reference/string/string/> or any other resource at <http://www.cplusplus.com/reference>. Finally, you may use the textbook and course slides. **You may not consult any other sources on the web, and you may not reuse code written before, whether written by yourself or any other person.**

Because you have from 9:30am to 11:59 on 3/3/2016, we will need to work on the honor system. You must put the following statement at the top of any source files you write, as part of your comment headers:

*I affirm that all code given below was written solely by me, <give your name>, and that any help I received adhered to the rules stated for this exam.*

Save your solution to GitHub, and add Matt as a collaborator. Matt’s GitHub ID is MTBellAtWhitworth.

**Program Specifications:**

You will be writing a program that supports utilizing an event venue. There are two classes to represent (see attached UML diagram). The **Venue** class has an array of **Event** objects as a property, and supports operations to add and look up events scheduled for the venue. Events should not be added to the venue’s schedule if the venue is fully booked; likewise events should only be added at available times. No two events may occur at the same time. If a proposed event cannot be added, print a message to the screen indicating failure. The **Venue** class should have a Boolean method named validTime() that tests to see if a proposed time for an event is already taken and, if it is, returns false. Events can be looked up either by time or by their name. Finally, there is a “default” event that represents free time: it has the name “free” and the time -1. If you try to look up an event and it isn’t there, the default event is returned.

Your **Venue** class should be defined in a file called Venue.h using inclusion guard. Its functionality should be implemented in a corresponding Venue.cpp file. Likewise, **Event** should be defined in a file called Event.h using inclusion guard, and event functionality should be implemented in Event.cpp. You will test your code using a main.cpp (see below).

**Testing your code:**

Save off the following in main.cpp:

#include <iostream>

#include <string>

#include "Event.h"

#include "Venue.h"

using namespace std;

int main()

{

Venue theSpot;

theSpot.addEvent(10, "Coffee Hour"); //Should work

theSpot.addEvent(11, "Brunch w/ Bob"); //Should work

theSpot.addEvent(11, "Bingo"); //Shouldn't work

cout << theSpot.findEvent(10).getTitle() << endl; //Should find Coffee Hour

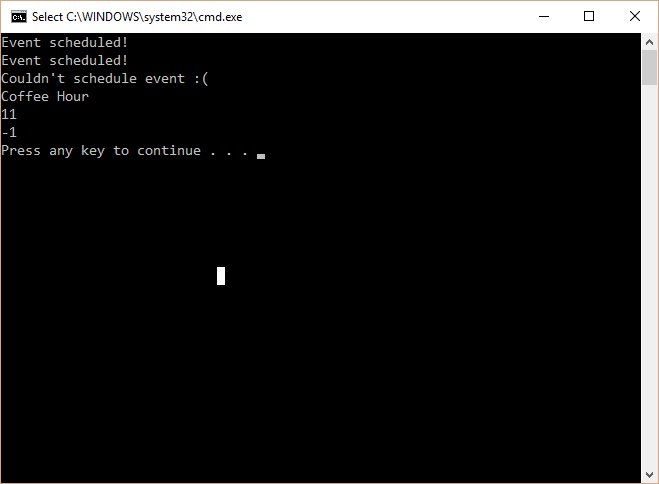
cout << theSpot.findEvent("Brunch w/ Bob").getTime() << endl; //Should find 11 o'clock

cout << theSpot.findEvent("Bingo").getTime() << endl; //Should print -1, because Bingo ain't there!

return 0;

}

Below is an example of what my screen looks like when I run this test code:



**Extra Credit Opportunity:**

Figure out how to do the above using the **vector** class so that you’re not limited to a max of 12 events. In order to get extra credit your code must work completely, so I recommend you first get the plain version working, save that off, then try your hand at the extra credit version. Only share one repository with me, either the plain version or the extra credit version.

**Other notes:**

This assignment should be do-able, but challenging! So, even though you can do it anywhere where you have network access, make sure to come to the classroom when the TA’s and I are there so that you can ask any questions and get other needed help. I’ll be in the lab with the room reserved for us from 9:30-10:50 on 3/3, and also from 12:50-2:10 on 3/3. Turn in something via GitHub by end of day, even if you don’t yet have a working program; that way, I can give partial credit and feedback.

Most of all, have fun and use this opportunity to learn!

Good luck ☺