Another Object Exercise

Since we are all still getting used to the idea of using objects and classes, here is another small exercise that involves the creation of a class. The Flight class that you create in this exercise will be reused in a later exercise that adds features and functionality.

For this exercise, you must create a class that is called Flight. The class has four methods, two getters for getMonth() and getDay(), one called set() and the last is called show(). The set() method takes six parameters for flight number (an integer), destination city (a string), departure time (a string), departure gate (a string), departure month (an int), and departure date (an int). The show method takes no parameters and outputs a string (to cout) that looks like the one that appears in the comment below. The message announces the flight number, destination, departure time, and gate, in a well formatted sentence. Both the set() and the show() methods return nothing (void). The class will need to have its own instance variables to remember the values from set() so that it can output the correct information when show() is called.

A complete .cpp file appears below (minus your file header comment section) to exercise your Flight class. In the code below, the main() function calls another function, called runTravel(). The runTravel function creates an array of 3 Flight objects, sets each with a different itinerary, and then calls show() on each flight object, in sequence. The array of Flight objects is named "leg."

This assignment is worth 10 points and is graded based on:

1. Creating a class in two files (.cpp and .h) that is used by the runTravel() function.
2. That you use the file below as is, with no changes.
3. That you create private instance variables in your class for the 6 pieces of information.
4. That you use the Project -> Add Class and leave the constructor and destructor as created.
5. That you use consistent and proper indenting and include a comment header in both the .h and .cpp files.
6. That your show() method outputs correct information for the Rome, Vienna, and Moscow Flight legs.

```
#include <iostream>
#include "Flight.h"
using namespace std;

void runTravel() {
    Flight leg[3];
    // flight #, destination, departure time, gate, month, date
    leg[0].set(225, "Rome", "8:00 am", "B12", 4, 21);
    leg[1].set(16, "Vienna", "11:30 am", "A5", 4, 21);
    leg[2].set(27, "Moscow", "4:15 pm", "C9", 4, 21);
    cout << "Here is your itinerary beginning on "
        << leg[0].getMonth() << "/" << leg[0].getDate() << ":\n";
    for (int i = 0; i < 3; i++ )
        leg[i].show();
    // Flight 225 for Rome leaving at 8:00 am from gate B12.
}

int main() {
    runTravel();
    return 0;
}
```