



# CST 280 - Program 1

Winter 2015  
Mon/Wed 8a-10a  
Instructor: T. Klingler

## Objective

To build a complete working C++ program to review basic C++ concepts focusing on multi-file C++ applications, input files, and strings.

## Overview & Instructions

Write a program to create a console application that allows the user to access sunrise/sunset and calendar almanac data for a given choice of cities. Your program should prompt the user for the following inputs:

- a date (one integer in the form `yyyymmdd`; for example **20150629** for June 29, 2015)
- a city's airport code (a three-character string; for example **MBS**)

Be sure to validate that:

- the date is valid
- the airport code is available (i.e. it matches on available from the given input file).

As your program launches, read the data from the file [cityinfo.txt](#) into one or more arrays. This will be the set of cities the use can choose from. Your program should first prompt the user for a city airport code (such as **MBS**). If this city is in the list of available cities, access the geographical coordinates (latitude, longitude) and timezone code (E-eastern, C-central, M-mountain, P-pacific).

Your program should calculate and write the following for this input:

- the calendar week within which the date exists (assuming a Sunday .... Saturday week)
- the sunrise and sunset for each day

It should include include a feature that will give the user a "Do you wish to continue" prompt to allow them to enter another city and/or date. Using one or more arrays for the city information will allow you to simply enter another array/list search instead of closing and reopening the input file.

For example, the input above should produce something like following output (sunrise/sunset values intentionally hidden). Be sure to include the day of the week as indicated.

SUN	28	JUN	2015	Rise:	5:58	am	Set	9:21	pm
MON	29	JUN	2015	Rise:	5:58	am	Set	9:21	pm
TUE	30	JUN	2015	Rise:	5:59	am	Set	9:21	pm
WED	01	JUL	2015	Rise:	5:59	am	Set	9:21	pm
THU	02	JUL	2015	Rise:	6:00	am	Set	9:21	pm
FRI	03	JUL	2015	Rise:	6:00	am	Set	9:21	pm
SAT	04	JUL	2015	Rise:	6:01	am	Set	9:21	pm

Include the correct day of the week name for the dates you include in the report (i.e. **MON**, **TUE**, etc.). Also include insure that all day numbers are two digits as shown in the example. Be sure that your date range correctly rolls into the next month and/or year, if necessary.

Finally add a feature that will automatically calculate if *daylight savings time* is active for the current date in focus. This is required as a parameter for the primary function used to calculate sunrise/sunset. At least one function in the provided calendar date function set can help with this determination. Daylight Savings Time (DST) in the United States:

- begins at 2:00 a.m. on the second Sunday of March
- ends at 2:00 a.m. on the first Sunday of November

Your program should utilize the following tools:

- File [cityinfo.txt](#): A list of city airport identifiers with geographical coordinates (latitude first, then longitude, then time zone).
- [sun.h](#) and [sun.cpp](#): Primary function for calculating sunrise and sunset from a date and geographical coordinate.
- [datefun.h](#) and [datefun.cpp](#): Several relevant date processing algorithms.

Utilize the "sun" and "date" functions in their current form - as external function sets that will be included into your main "driver" program for this application. You have formal permission to use all instructor-provided functions provided credit is given within comments in your code as as long as the provided functions are unchanged. Additionally, although c-strings have only been introduced so far in the course, you are free to utilize string objects instead for this solution.

Design your application using guidelines to maximize modularity, reusability, and maintainability.

---

## Deliverables

**Deliver** the following to the online course management system **dropbox** as your final product:

1. **Upload one PDF** document containing:
    - **Source code** files (.cpp and/or .h files; only those build/changed by you)
    - **Output** demonstrating the following tests cases:
      - MBS; 20150629 (validating the example above)
      - LAX; 20151228
      - DEN; 20151106
      - LAF 20150704
      - FWA; for any date (expecting an error message for city)
      - DFW; for 20130931 (expecting an error message for invalid date)
  2. **Upload** your **source code files** (.cpp and/or .h files) to the dropbox in the event your instructor wishes to execute your program
-