# Map Project

For this project we will use a data file from an Excel spreadsheet that contains the schedule of classes for the Summer 2022 session. Excel spreadsheets can be saved in a text format where the columns for each row are delimited by a comma. The extension for this type of file is .csv.

The first row in a .csv file typically contains the column headers for the spreadsheet. In this file the column headers are: Subject, Catalog, Section, Component, Session, MinUnits, Units, TotEnrl, CapEnrl, AcadOrg, Instructor, Capacity, Room, MtgStart, MtgEnd, Days, StartDate, EndDate, Term, Campus, ClassNbr, Total Credits, DUP, FULL, OVER.

For our program, we do not want to look at all these columns, so the columns we will take from the file will be:

Subject, Catalog, Section, Component, Session, Units, TotEnrl, CapEnrol, Instructor. The Units, TotEnrl, and CapEnrol fields are integers, all the other fields are strings. We will look at how to read a file like this in lecture.

Here is an example of a few lines from the file after it has been read into memory and placed in the map structure:

**Subject Catalog Section Component Session Units TotEnrl CapEnrl Instructor**

**AUT 101 50H LEC 10W 3 5 24 Broadley Jr.,Harold Z**

**AUT 236 01H LEC 10W 4 5 24 Burge,Mark D**

**BIO 101 01H LEC 8W1 4 5 24 Staff,Tba**

**BIO 101 01HL LAB 8W1 0 5 24 Staff,Tba**

**BIO 101 01W LEC 8W1 4 5 24 Staff,Tba**

**BIO 101 01WL LAB 8W1 0 5 24 Staff,Tba**

**BIO 101 50H LEC 8W1 4 2 24 Staff,Tba**

**BIO 101 50HL LAB 8W1 0 2 24 Staff,Tba**

**BIO 101 D01 DED 8W1 4 21 30 Martin,Jennifer M**

**BIO 101 D01L LAB 8W1 0 21 30 Martin,Jennifer M**

**BIO 101 D02 DED 8W1 4 22 30 Song,Pengfei**

**BIO 101 D02L LAB 8W1 0 22 30 Song,Pengfei**

**BIO 101 D03 DED 8W1 4 3 30 Martin,Jennifer M**

**BIO 101 D03L LAB 8W1 0 3 30 Martin,Jennifer M**

**BIO 102 20H LEC 8W1 4 5 24 Staff,Tba**

**BIO 102 20HL LAB 8W1 0 5 24 Staff,Tba**

**BIO 102 D01 DED 8W1 4 20 30 Evans,Ann S**

**BIO 102 D01L LAB 8W1 0 20 30 Evans,Ann S**

**BIO 141 01W LEC 8W1 4 6 24 Marcinkus,Susan Rita**

**BIO 141 01WL LAB 8W1 0 6 24 Marcinkus,Susan Rita**

**BIO 141 20H LEC 5W1 4 0 24 Staff,Tba**

The key for each record will be the combination of the Subject, Catalog, and Section. For example, the key for the first record would be AUT\_101\_50H. Using these three fields guarantees that the keys will be unique.

First you will write a class that represents a row in this data structure (I called it **scheduleItem**, but you can call it whatever you want). Each of the fields in the report will be an instance variable. You do not need to have a **set** function for these variables, but you will need **get** functions for each of these.

In addition, you will want to overload the **==**, **!=**, and **>=** operators so that they will return true if the key fields for the operands match.

The **print** function should write out the data for the instance variables (a single row) in a format as shown above.

Write the constructor for the class so that you can initialize all the fields using default parameters. The initial values for the string fields can be the empty string **“”** and for the integer files can be **0**.

The second class will be the container class for this data. The instance variable in this class will be a map variable with the keys being the same as the key described above and the items in the container being objects of the **scheduleItem** class (I called this class **schedule**, but you can call it whatever you want).

The first operations for this class will be a member function called **initSchedule** which will take the **ifstream** pointer to the data file as input (open the data file in the client program). This function will read and discard the first record (which contains the column headers) and then will read the required fields from each row. It will create a **scheduleItem** object using the data read from the file and then pass that object to the **addEntry** function which will insert the object into the map. This is very similar to what we did with the Address Book project.

The **schedule** class should have a **print** function that uses the iterator to read and print all the records from the file to produce the report shown above. You can optionally have a **printHeader** function that prints the column headings.

We also want to implement three find functions: one that finds and prints all the records with a target subject, a second to find and print all the records with a target subject and catalog number, and a third which finds and prints all the records given the instructor’s last name.

You will need the following header files (in addition to the usual) for this class: **map**, **iterator**, and **utility** (this is for the **pair** function).

Finally, you will write a client program that declares a **schedule** variable, opens the data file and calls the **initSchedule** member function, then sets up a menu loop which allows the user to select one of the find functions and to print the schedule.

Write a report with the answers to these questions and submit it to Canvas when you are ready for me to grade this project:

Reflection Questions:

1. What did you find most challenging with this program?
2. What problems did you encounter and how did you solve them?
3. What did you learn from writing this program?