Program 2

Introduction to C
CSCI 112, Fall 2020

Objectives

- Use structs
- Read from a file
- Create a header file
- Work with character arrays, equivalent to char*
- Use a makefile to compile multiple source files

Description:

Create a program that reads in a list of CSCI classes from a file. Store the list in an array of data structures. Allow the user (a student) to have the following options:

1) Print a class given a CSCI number
2) Print all classes available on MWF or available on TR.
3) Print all classes taught given by a particular professor's last name
4) Quit

All output will be written to the screen.

Requirements:

You must have an array of structs that stores all the data for the classes. This struct will have the ID of the class (CSCI 112), title of the class, the days of the week the class is offered, the time of the class, how many seats are available, and the professor teaching the course.

You must use the input file in /public/pgm2/classes.csv to input the data.

You must provide the options to the user on the screen as shown in the screen shot below.

You must provide the choices (CSCI number, professor's name, etc) to the user after they have made a selection for which option they want. The user will then provide the choice. You must write all information asked for by the user to the screen (stdout).

To print class information, print in the following order:

Title, Class number (ex: CSCI 112), Taught by, number of seats, day, time.

You must loop on user queries until the user selects the quit option.

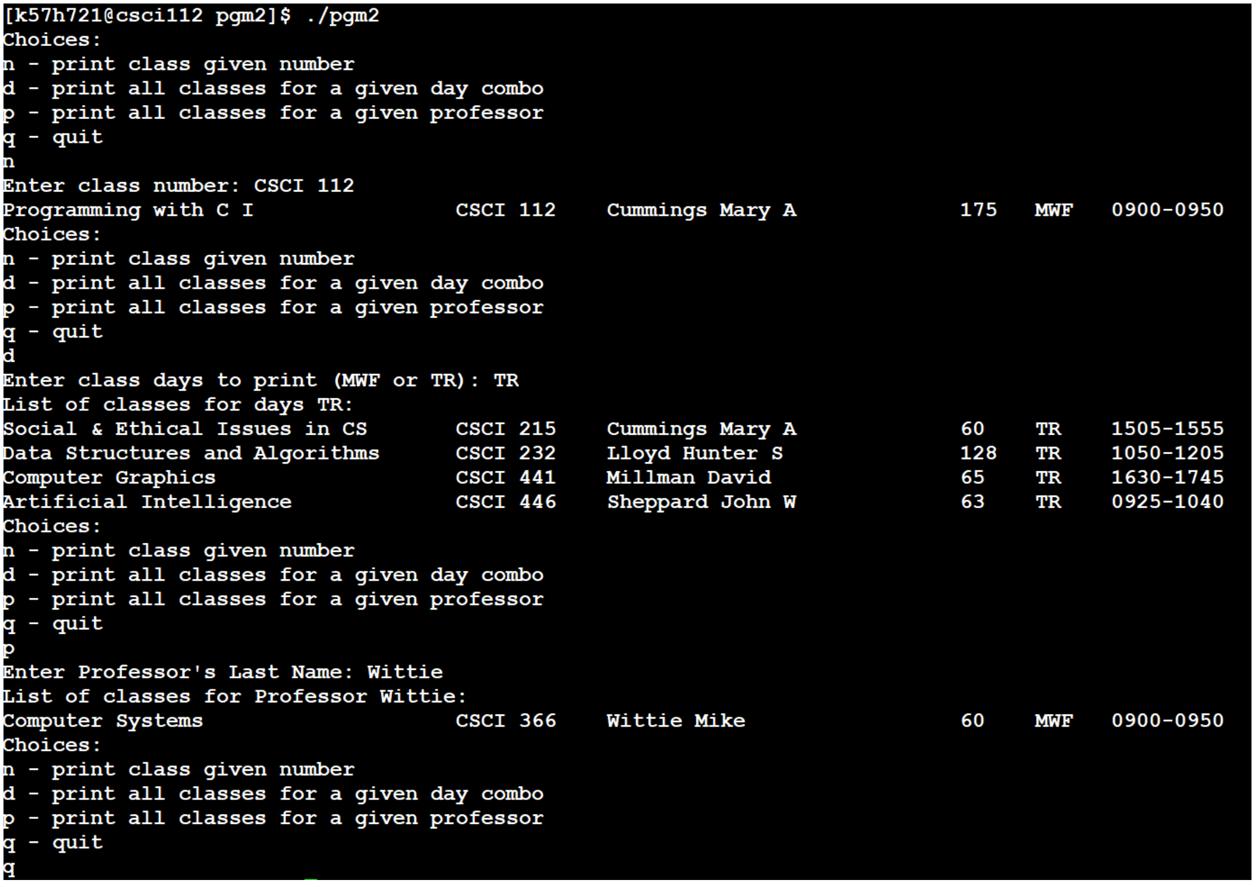
Place each function or group of functions in a separate file (.c) and have a makefile that compiles all your code together to make the executable. A group of functions means that you might have all your print functions in one file, but you would not have the query function in the same file as the print functions.

You must put your struct in a header file that you create.

NO GLOBALS.

You must make 1 run – using all 3 options. print class given number, please classes for a given professor, print classes for a given day combo

My Output:



```
[k57h721@csci112 pgm2]$ ./pgm2
Choices:
n - print class given number
d - print all classes for a given day combo
p - print all classes for a given professor
q - quit
n
Enter class number: CSCI 112
Programming with C I                CSCI 112    Cummings Mary A             175   MWF   0900-0950
Choices:
n - print class given number
d - print all classes for a given day combo
p - print all classes for a given professor
q - quit
d
Enter class days to print (MWF or TR): TR
List of classes for days TR:
Social & Ethical Issues in CS       CSCI 215    Cummings Mary A             60    TR    1505-1555
Data Structures and Algorithms      CSCI 232    Lloyd Hunter S              128   TR    1050-1205
Computer Graphics                   CSCI 441    Millman David               65    TR    1630-1745
Artificial Intelligence             CSCI 446    Sheppard John W             63    TR    0925-1040
Choices:
n - print class given number
d - print all classes for a given day combo
p - print all classes for a given professor
q - quit
p
Enter Professor's Last Name: Wittie
List of classes for Professor Wittie:
Computer Systems                    CSCI 366    Wittie Mike                 60    MWF   0900-0950
Choices:
n - print class given number
d - print all classes for a given day combo
p - print all classes for a given professor
q - quit
q
```

Submission

- Due Date: 11/4 at 11:00pm

  Each student will complete and submit this assignment individually. I will check for plagerism. Labs submitted after the due date/time will not be accepted.

Grading

Points (100 pts) – IF YOU DO IT ALL CORRRECTLY BUT ONLY USE ONE SOURCE AND NO MAKEFILE YOU WILL GET A 90

- 5 points – comments explaining what your program does
- 10 points – submit source code
- 10 points – submit screenshot showing successful compile of all source files (do a touch *.c first) and run
- 15 points – compiles successfully with -Wall – no warnings
- 5 points – made one run with the 3 options and quit
- 5 points – used and submitted a makefile
- 5 points – had each function or group of functions in a separate file
- 5 points – read from the specified input file
- 10 points – printed the information specified in the requirements
- 10 points – created an array of structs
- 10 points – has the required options presented to the user
- 5 points – loop on queries
- 5 points – put the struct in a header file