CS1050 – Lab 13

Fall 2021

Concepts to Practice

- Structures
- Arrays of structures

Submission Information

Submit this assignment by following the instructions given by your TA. SUBMIT ONLY the .c file (no a.out or executable file is required). All of the lab assignments must be submitted before the end of the lab using the lab code given by the TA.

Use the following submit command:

```
mucs submit <class> <assignment_name> <filename>
```

For example:

```
mucs submit 1050 lab13 lab13.c
```

Description

To get started on this lab, type the following while logged in to tc.rnet.missouri.edu:

```
cs1050start lab13
```

For the lab assignment, you are to start with the starter code provided in the lab13 directory. This starter code is an outline of what you need to do. The main() function is done! So, don't change main(). Instead, your job is to fill out the following provided functions:

- GetCourseByDeptAndNumber()
- GetCourseByID()
- PrintCourse()
- And for bonus points: GetStudentsByLastName() (and GetStudentsByFirstAndLastName() for the Honors section)

There is base code for these functions already in the lab13.c that you are provided. However, this base code doesn't do anything. It is up to you to create functions that work properly.

You are provided with a library just as you were in the prelab. The functions are in a provided header file called university.h and a library that will automatically be linked in. The functions I have provided give information about MU classes and students enrolled in some of those classes. Just like printf(), scanf(), or other Standard C Library functions you have worked with in the past, you do not have the source code for the functions I have provided.

You can look at the provided university.h header file to see the different structures defined (Course and Student) as well as various provided functions. One convention that my library uses is that the last Course in an array of courses should have an id of -1 (as a sentinel value to indicate the end of the array). Similarly, the last Student in an array of students should have a student number of -1 (as a sentinel value to indicate the end of the array).

You also have a Makefile that you can use to build the lab. This time, the executable file is called muapitest (so just type ./muapitest to run it). So, to get started:

- cs1050start lab13
- cd lab13
- make
- ./muapitest

At this point, you can start modifying lab13.c to implement the functions. If you aren't sure how a function should work, just look at the tests in the main() function. For example, calling GetCourseByID(5) should return the course that has an id of 5 (which, you will notice in the sample output is "Introduction to Logic Systems").

Hints

- You can use the other functions in university.h.
- If you want to use strcmp(), you need to #include <string.h>.

Honors

Honors students do not have anything extra on this lab, but if you want all of the bonus points you must implement both GetStudentsByLastName() and GetStudentsByFirstAndLastName().

Bonus

Implement the bonus function called GetStudentsByLastName() to get bonus points. Consider that this returns a pointer just like the function GetAllStudents() does. How will you create such a thing? Could you use malloc()? Could you use a static variable?

Sample Output (output from just the starter code)

```
jimr@JimRXPS13:~/CS1050/FS2021/labs$ cs1050start lab13
Cloning into 'lab13'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 6.09 KiB | 1.52 MiB/s, done.
jimr@JimRXPS13:~/CS1050/FS2021/labs$ cd lab13
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab13$ make
gcc -std=c11 -g -Wall -Werror -c lab13.c
gcc -std=c11 -g -Wall -Werror lab13.o -luniversity -L. -o muapitest
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab13$ ./muapitest
Test #1:

Test #2:

Test #3:

Test #4:

Test #5:

Test #6:

Test #7:

Test #8:

Test #9:

Test #10:

Test #11:
```

Non-honors Sample Output (after creating your solution)

```
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab13$ make
gcc -std=c11 -g -Wall -Werror -c lab13.c
gcc -std=c11 -g -Wall -Werror lab13.o -luniversity -L. -o muapitest
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab13$ ./muapitest
Test #1:
ID=5,Name=Introduction to Logic Systems,Dept=CS,Number=2270,Instructor=Yunxin Zhao
Test #2:
ID=11,Name=Introduction to Probability and Statistics II,Dept=STAT,Number=3500,Instructor=Teri Christiansen
Test #3:

Test #4:
ID=2,Name=Algorithm Design and Programming I,Dept=CS,Number=1050,Instructor=James Ries
Test #5:
ID=12,Name=Experimental Design,Dept=STAT,Number=4540,Instructor=Isabella Zaniletti
Test #6:

Test #7:

Test #8:
1 - Abbie Ries
2 - Charlotte Ries
3 - Allison Ries

Test #9:
27 - Bob Walkenhorst

Test #10:

Test #11:
```

Honors Sample Output (after creating your solution)

```
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab13$ make
gcc -std=c11 -g -Wall -Werror -c lab13.c
gcc -std=c11 -g -Wall -Werror lab13.o -luniversity -L. -o muapitest
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab13$ ./muapitest
Test #1:
ID=5,Name=Introduction to Logic Systems,Dept=CS,Number=2270,Instructor=Yunxin Zhao
Test #2:
ID=11,Name=Introduction to Probability and Statistics II,Dept=STAT,Number=3500,Instructor=Teri Christiansen
Test #3:

Test #4:
ID=2,Name=Algorithm Design and Programming I,Dept=CS,Number=1050,Instructor=James Ries
Test #5:
ID=12,Name=Experimental Design,Dept=STAT,Number=4540,Instructor=Isabella Zaniletti
Test #6:

Test #7:

Test #8:
1 - Abbie Ries
2 - Charlotte Ries
3 - Allison Ries

Test #9:
27 - Bob Walkenhorst

Test #10:

Test #11:
2 - Charlotte Ries
```

Guidelines for Grading Lab 13
40 Points Possible (+6 bonus points)

General

If your program does not compile or produce any input/output (I/O) because most of the source code is commented out then your lab will receive a grade of ZERO POINTS. Further, if your program does not actually follow the specifications, but merely prints out lines that make it appear to follow the specifications, you will receive a grade of ZERO POINTS. For partial credit your C program must not only compile but also produce some valid I/O that meets the lab specifications.

You program is expected to have a comment header at the top that includes your name, pawprint, the course you are taking, and the lab that you are solved (e.g., "Lab 13"). Your code should be nicely indented. You may not use global variables. **You will lose up to 10 points if you do not meet these basic requirements**.

5 points: Your code makes absolutely no changes to main().
5 points: Test #1 works as shown.
5 points: Test #2 works as shown.
5 points: Test #3 works as shown.
5 points: Test #4 works as shown.
5 points: Test #5 works as shown.
5 points: Test #6 works as shown.
5 points: Test #7 works as shown.

Non-honors BONUS (up to 6 points)

2 points: Test #8 works as shown.
2 points: Test #9 works as shown.
2 points: Test #10 works as shown.

Honors BONUS (up to 6 points)

2 points: Test #8 works as shown.
1 points: Test #9 works as shown.
1 points: Test #10 works as shown.
2 points: Test #11 works as shown.