Lab 1: Review of structured, looping and modular programming

David E. Wells, Jacob Rush, Steven A. Savedge

CS-230 Data Structures

# Introduction

The materials used in making this lab include: Windows 10, C++ lde(Dev-Cpp 5.11 TDM-GCC 4.9.2), <https://github.com/TheTrueMalkavian/CS-230_Lab1>. In this lab we where tasked with creating a program that allows the user to initially enter information on 4 students and 10 courses. The program then had to associate each student to at least four courses for a semester session. We also had three specific requirements in option 3 shouldn't work if you have no students and courses in the database, Option 1 forces the user to enter a minimum of 4 students, Option 2 forces the user to enter a minimum of 10 courses.

## Procedure/Methodology

*David Wells*

I started to look back at my old notes from CS-200 to try and apply what I learned from the banking app to use with this program. I ended up getting a little confused and decided to just take everything that we went over in class and rewrite it for use in this project. I started by writing in the menu and then filled out students header file. After this I filled in everything for the .cpp file, all while regularly compiling the code. The first error I ran across was when I made the Courses.cpp file I ended up included the .ccp file itself and it got caught in and endless loop. Then I found that I didn’t write down the functions in the header file that I wrote into the .ccp file. I ended up getting stuck at option 3 and trying to figure out how to get the function to work. I realized one way was that I could make a nested switch statement and put their choices into another array.

*Jacob Rush*

First, I had to look at some of the applications that we had built in class to create a base for the application. This ended up working for the creation on the project, but it didn’t cover a lot of the core functions of this application. The main issue I kept running into was the correlation of the courses to the student. We ended up having to create a for loop that incremented through the courses to assign them on the output.

*Steven Savedge*

I started by creating the classes for student and course. Beginning with the header files I declared the basic data the classes would need and created accessors and mutators for them. I decided that the student class couldn’t hold class objects, because that would be redundant from the main class array. So, the student class would instead hold an array of class pointers that point the classes they have. I then wrote the accessors and mutators and moved onto the main method. The main method initializes an array for both the students and courses, and two integers for the length of each. It then calls a menu to prompt the user and uses a switch statement to process the input. The first menu option, Figure 1., was to enter a minimum 4 students prompting the user for how many to be entered then asks for the user id, first name, and last name checking to see if another student in the array already has that id. The second menu option is like the first, it asks for a minimum of 10 courses then for their course id, course name, and how many credits it is, also checking to see if the id is a duplicate, Figure 2. The third option is to assign at least 4 courses to a student. First it asks for a student id until the user inputs a valid user id. Then asks for how many courses to be inputted. And finally takes course ids from the user and checking if they are valid to be assigned to the student. The fourth menu option, *Figure 4*, just simply loops over the array of students and calls the students print method to print all their data. The final most important option is an even more simple exit option, which just ends the switch loop. Most of the errors I ran into were easy to fix syntax errors. One error that I did run into is with the include statements and the class course being defined multiple times. That’s where I looked online and learned about include guards the check to see if the class has already been loaded to prevent it from being loaded multiple times.

#### Analysis.

*David Wells*

I liked the way that cs 200 gave us time to get used to git hub. I found that all my teammates ended up having to make an account and were unfamiliar with it. It even took me a while to get used to it. Last semester I couldn’t get the GUI to work and ended up using the command line for everything. This semester I was struggling with the command line and easily got the GUI to work. While I was working in the Cyber Lab I found that our group wasn’t the only ones with bad communication. On Thursday we all got on discord and github but Friday we ended up picking Jacob onto our team. I think this also shows that we didn’t have much communication. I believe it might be the way that room is compared to us being in square desks where we can face each other and talk amongst our selves better. I believe the class in the library is great for lectures, but not for gathering groups or any type of group work. Also as a result of me and my teams poor communication and tardiness. I didn’t realize I should have explained how git hub works and that we all should be working on the same code. The result was that now we pretty much have 2 branches of different code.

*Jacob Rush*

This lab was a learning experience for me. I had to first pull from the knowledge of my cs-200 class to try and write this program. Our group also used github to collaborate on the code. This was my first time using github so it made it a big learning curve for me. But after figuring out how it works, I was able to efficiently contribute to the lab.

*Steven Savedge*

I had begun coding the project on the 17th and didn’t even realize it was a group project. In class on the 19th I was told to join the groups git hub. I hadn’t much used git hub before, but the concept was easy to handle. The code that they had was built from the examples in class and was quite different from mine. So, I created a branch from their master on git hub to hold my code and be used to help some of the problems we were solving.

**Conclusion**

*David Wells*

This project I am starting to feel more comfortable with C++ but find that I come to roadblocks and it takes me about a day to kind of realize how to do it. Overall this project was very rushed, and this is mostly my fault as I should have made sure on Tuesday everyone knew what they were doing. Thursday came along and I found out that no one has even heard of git let alone started working on the project. I think group projects are great and that I really need to work on my communication skills.

*Jacob Rush*

This lab made me look back at CS-200 to remember some of the skills I had forgotten to be able to create this project. Due to this I know I have some brushing up to do on my coding technique in C++. I also got to learn a valuable skill in learning how to use github. This skill will come in handy when I go to collaborate with other coders.

*Steven Savedge*

This project has helped me catch back up with C++ and object-oriented programming. I now have git hub set up and understand how to connect with it. I feel like this project could have gone a lot better and more smoothly if everyone had started on the same page. Instead I was unaware of the format for the first lab, but I will be ready for the next.

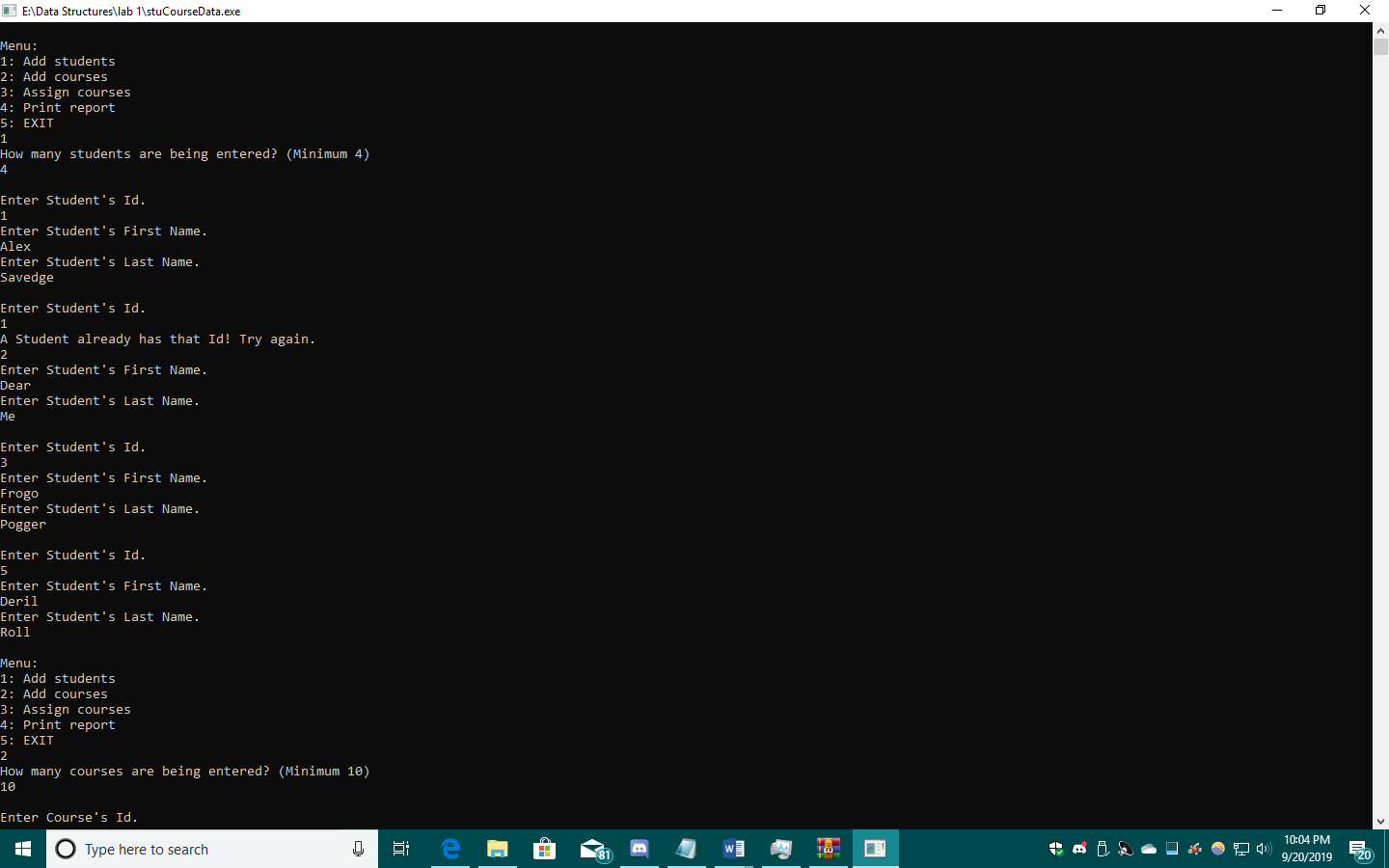
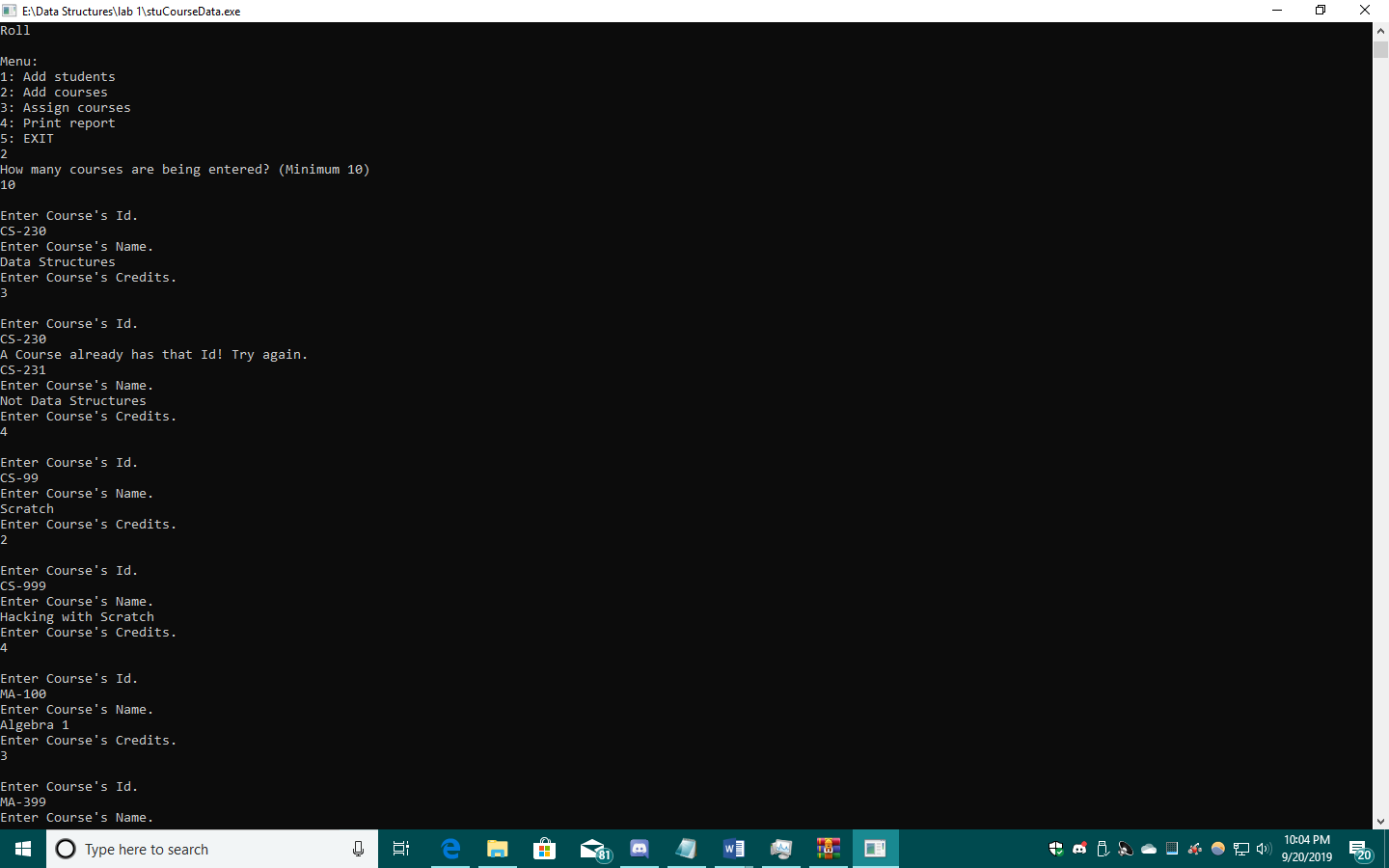
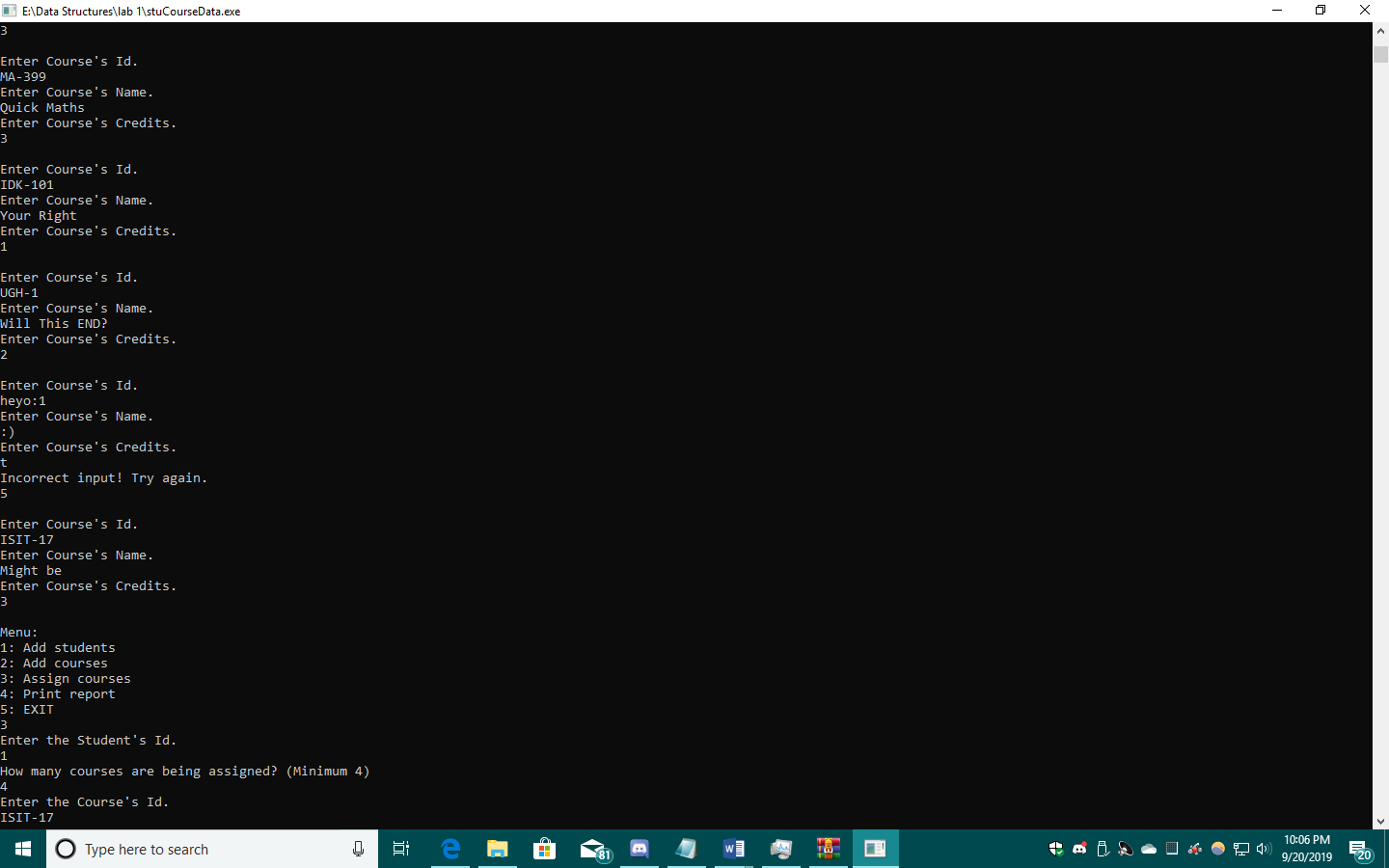
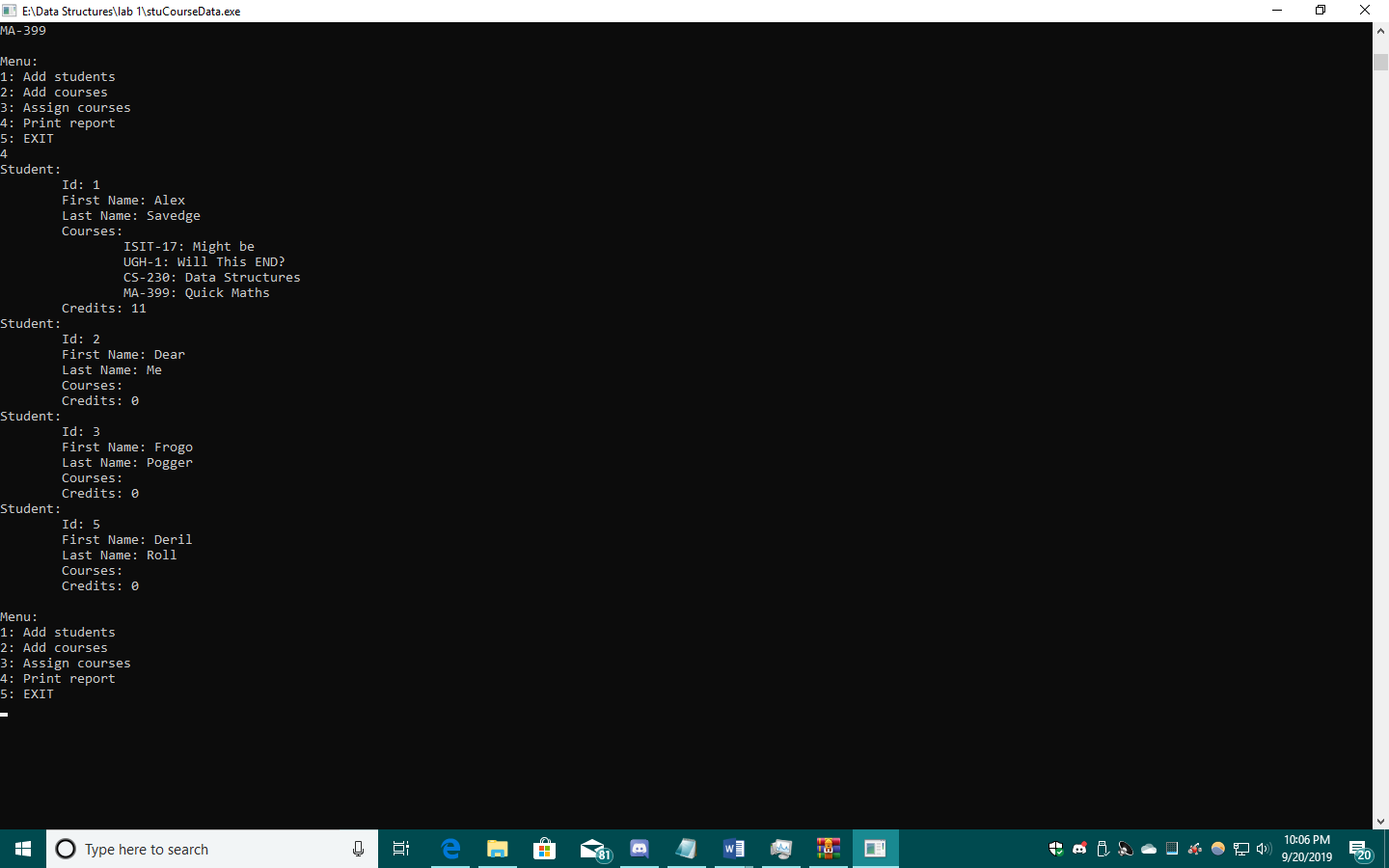
References

Professor Andrew Mehri’s class files

David Wells C++ class notes

Include guard. (2019, July 24). Retrieved from https://en.wikipedia.org/wiki/Include\_guard

Figures

 Figure 1.  Figure 2.  Figure 3.  Figure 4.