COSC 220 - Project 1 Evaluations

Dr. Joseph Anderson

[jtanderson@salisbury.edu](mailto:jtanderson@salisbury.edu)

Due: 22 October 2019

Name: Devin Schmidt

Name of peer: Cody Murrer

Objectives

1. Practice peer code review

2. See how peers implement different features

Task

Find another person in the course to work with for this lab.For this lab you will worktogether to evaluate and improve your project 1 submission. Peer code review and debugging is animportant skill in the workforce !

In addition to any bugs found and fixed, work with that person to add documentation to theircode, clearly explaining 1) the overall code structure, 2) the function and purpose of individualmethods/subroutines, and 3) potential areas for improvement : code refactorization, features to beadded, and possible future problems. Some basic “checklist” questions you can ask yourself are :

1. Am I able to understand the code easily ? The code was easily understandable.

2. Is the code written following good coding standards/guidelines ? I think good coding standards and guidelines were used.

3. Is the same code duplicated more than twice ? Yes, in main input validation is repeated more than twice.

4. Can I test / debug the code easily to find the root cause ?

5. Is this function or class too big ? If yes, is the function or class having too many responsibilities ? The code seems to be a reasonable size.

6. For important objects in the code, does ownership change often ? ??????????

Is the logic easy to follow ? Yes, the logic is straight forward.

Compile and run their code for Project 1, then complete the following evaluation tasks.

Perform the following tasks :

1. Add at least three students, with all appropriate metadata.

2. Print the list of students to verify they are all there.

3. After adding the students, change the name of one of them.

4. Print the student to make sure the name changed.

5. Add at least one course onto each student.

6. Print the full list of students with courses to make sure the courses worked.

7. Add at least three courses onto one student.

8. Print the full list of courses again to verify.

9. Delete one of the students.

10. Print the database again.

11. Delete one of the courses from one student.

12. Print the details for that student.Save the output of the above procedures and print it out to be turned in.After you have tried using the program, read through the source code to see how their programworks, and answer the relevant questions about it below.Finally, work together to improve or fix one of the major issues or deficiencies with the program.

Submission

You will turn inyourownProject 1 if you made changes to it. The grade of this lab will be tosee if you made relevant changes and bug fixes, added documentation, and completed a thoroughreview of your partner.

Code Testing and Evaluation

Run the program. Try different inputs. Describe any bugs or errors you encountered :

* When user inputs data they must press enter again to see the next line about to be printed until then the cursor rests on a black line
* Course field is involuntarily skipped when user is filling out required data and the next field after that is then ready to be entered
* When the field is skipped the courses do not print
* I also do not have an option to add courses to more than one student

Does the program satisfy all the features required by the project instructions? If not, why? There are some missing requirements. There is not an update or delete. The code is not written for it.

Was the program easy to use?

The program was easy to use.

Did you have to ask for help?

No

What parts of using the program did you enjoy?

I liked how the program is straight forward and has some input validation to make it user friendly.

What parts of using the program did you not enjoy?

I do not like how the procedural the program is. I would like to go back to adding courses or majors if I wanted to.

Read the program source code. Can you identify where any bugs occur in the program? There might be a misplaced cin ignore somewhere or a missing one. The cins for courses don’t use getline so white space is not recorded. Which could solve most of the bugs. The option to add more courses to another student can be added in when the user says no to entering more courses on the current student.

Read the program source code. Describe the algorithms used. Does the logic of the program make sense to you? Can you identify parts that you would change to make it simpler? Are there “clever” aspects of the implementation that you haven’t seen, but which you would use later?

The program was made “runnable” by using while loops with a user input case. Cins stored data in variables to get passed into a method for use. While loops wrapped some user input options for input validation. The student DB and student class have the same functions of a basic linked list. Course is a basic class for setting and retrieving data. Main passes data to all these classes to create the Database esque structure.

Read the program source code. Is it clearly documented?

No.

Would you be able to easily take the code and modify it to, for example, complete one of the bonus tasks if it is not already done?

I would be able to modify it since it has the required structures for those specific tasks.

Identify specific areas where extra documentation is needed. Discuss with your partner these needs and make notes for them to add the documentation on their own. He should document the main. Adding comments to specify which part of main is doing what and comments describing what each function does in StudentDB and Student.

Is there any aspect of your own program (user interface, code organization, control flow) that you would consider changing in a “version 2.0” after seeing this person’s implementation?

Any other general comments about the code you are reviewing? Makes me contemplate which bracket style to use.