NARRATIVE FOR ENHANCEMENTS 1, 2, & 3

ALL PERFORMED ON: ClientList program

ARTIFACT HISTORY

My artifact for the: 1) Software Design and Engineering, 2) Algorithms and Data Structures, and 3) Databases, categories is a program titled, “ClientList”. This is a project from my CS410 class, Software Reverse Engineering. This is a simple program that prompts the user to select from a menu of options and then reacts according to that input. The program will display a current list of clients, and then allow the user to select a specific client to view their investment option or exit the program. I chose this program because it seemed like a great example of a real-world scenario and could allow for expandability. I imagined this type of program to be much like one that a business such as a bank, for example, may use to keep track of client information. The program very adequately showcases my ability to transform a real-world business need and into a simple and sophisticated software solution.

ENHANCEMENTS

For this artifact I performed the following enhancements to align with the three different categories mentioned above:

1) Software Design and Engineering

   -For this category enhancement, I converted my original C++ program into Java. Being able to translate a program from one language to another was a great way for me to showcase my computer science skills for this category, and a valuable skill in the ever-evolving development world.

2) Algorithms and Data Structures

-For this category enhancement, I expanded the complexity of the original program by converting the list of clients from individual variables to members of an ArrayList. This showcased skills and knowledge of data structures as well as made the program more efficient and easier to expand in the future. For this category I also implemented some improvements regarding the security of the program by adding input validation and exception handling. Now the program will verify that the user input is within the range of the menu options. If not, the user is prompted with a message to make a valid entry. The user input required for the program to run must be an integer. In order to provide further input validation, I used try/catch blocks to handle any exceptions thrown from a letter being entered instead of an integer.

3) Databases

-For this category enhancement, I chose to use the ArrayList built by the program, to populate a MySQL database. This was a great way to solve a real-world business problem and allow me to demonstrate my ability to work with databases in a way that incorporates useful business tools and software solutions. Now, every time the program is exited, a connection to a MySQL database is established that loads the contents of the clientList ArrayList into a database.

REFLECTION

I learned quite a bit from revisiting my original program and then performing the above enhancements. I was reminded of the similarities of C++ and Java. Not surprising since Java was built on a C++ foundation. However, there were some syntax differences like method naming for instance that I had to correct. I also had to go about input validation a little differently than I would have had to in C++. These challenges, however, only strengthened my programming skills and led to a higher level of confidence and better performing program. Feedback was tremendously helpful as I navigated my way through the enhancements. I learned from that feedback that my original plan for the Databases category was not strong enough to showcase my proficiency in this area. I wanted to simply create a user log-in database for the program. However, the program now is much more sophisticated in the Databases category after building upon professor feedback. Instead, the program will load a MySQL database upon closing the program with the contents of the ArrayList of client names.

MEETING OUTCOMES

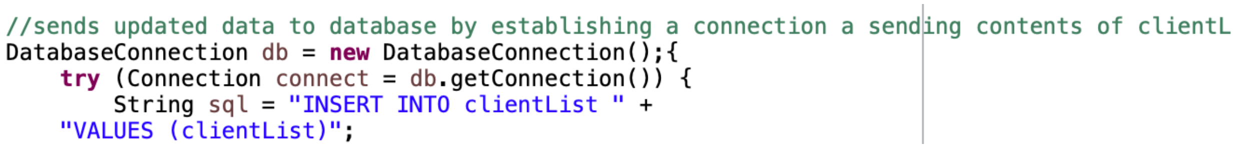
The program described above allowed me to demonstrate my proficiency in a range of computer science disciplines. It was also a great learning opportunity to revisit an old program and make necessary enhancements to it. Along with these advantages, this program, and the work that I performed throughout all three categories of enhancements, clearly lay out my successful completion of the *Five Course Outcomes* outlined below:

1) **Employ strategies for building collaborative environments that enable diverse audiences to support organizational decision making in the field of computer science**

CS499_ClientList/src/ClientList.java - Eclipse IDE

i. I was able to achieve this in my portfolio with my artifact, ClientList, by taking a program originally written in the C++ language and translating it into Java, while maintaining its efficiency and honoring its original design features. This process is proof of my ability to operate in a collaborative way and be able to adapt to different audiences, diverse programming environments, and make decisions necessary while adhering to design techniques and user requirements. Java is the main language used by Android mobile devices and with the growing list of available devices, and greater and greater demand for mobile platforms for programs, being able to convert a standard desktop-style C++ program into Java is a very valuable and meaningful skill.

ii. I also successfully showcased my proficiency in this course outcome by my enhancement in the Databases category with this artifact. This enhancement meets that criterion by its utilization of trusted database software, MySQL, and taking advantage of its cohesiveness with Java and the Eclipse development environment. By using these combined tools, I have shown that I am able to use well-founded and innovative techniques to solve a computing problem. First, I chose to use the ArrayList of clients I built for enhancement two to automatically populate a MySQL database. Second, I also created a DatabaseConnection class where I wrote out individual SQL queries that not only portray my experience with SQL databases, but also serves as another way to populate a database from a Java program. The solution offered in this program is an industry-specific one that will meet many relevant business needs: collecting client information, verifying that information, protecting that information, and allowing for expansion in an easy to maintain and simple software solution.



```
//sends updated data to database by establishing a connection a sending contents of clientL
DatabaseConnection db = new DatabaseConnection();{
    try (Connection connect = db.getConnection()) {
        String sql = "INSERT INTO clientList " +
    "VALUES (clientList)";
```

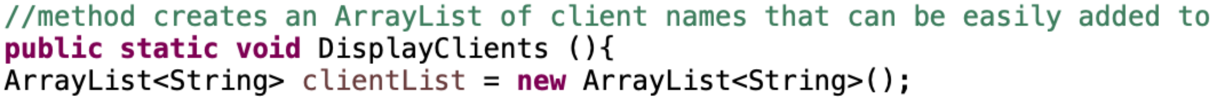
2) Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts

CODE REVIEW

i. I was able to meet this course outcome in my portfolio by performing a code review of my original version of my artifact, ClientList. For this code review, I created a screencast where I was able to walk through an evaluation of the program and highlight areas for improvement. The video can be found on YouTube by clicking here. This code review is an example of the skills I have obtained in analyzing a program, identifying its structure, and finding room for improvement. This code review covers an analysis of the structure, documentation, variables, arithmetic operations, loops and branches, and defensive programming techniques. Code reviews are a regular part of a software developer’s duties. The code review included in my portfolio showcases my knowledge of programming basics and best practices. It also shows my ability to create visuals and presentations that are easy to operate, easy to follow, and professional.

ii. I was also able to meet this course outcome by including descriptive, informative, and useful in-line comments to the code in my artifact, ClientList. The comments included in this program describe its functionality and purpose in a way that will make it easy for any developer to follow. This skill is one that is vital for a programmer to have. Most projects are done in a collaborative way, and it is very important that the work that one developer does to a program can be easily deciphered by another developer. Having people who are proficient in this area of the software development world is the only way we can have the sophisticated software solutions available to us today.

3) Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution, while managing the trade-offs involved in design choices

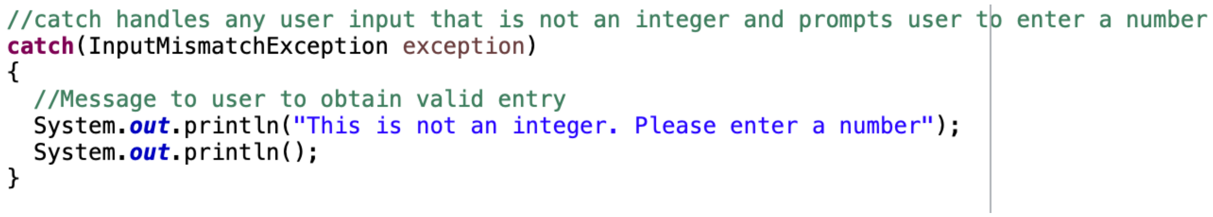


```
//method creates an ArrayList of client names that can be easily added to
public static void DisplayClients (){
ArrayList<String> clientList = new ArrayList<String>();
```

i. I was able to achieve this outcome by the enhancements I made to my ClientList artifact in category #2. For this enhancement I took the names of the clients and created an ArrayList data structure to house the list of clients. My original artifact displayed a list of client names by simply creating a separate variable for each client. This enhancement now much better aligns with appropriate business needs and software solutions. In the real world, a business that would use a program similar to this would have a list of clients that is fluid. It would need to be able to grow and change with the business. An ArrayList is a more advanced programming concept that allows for expandability and better showcases my programming knowledge. This enhancement shows that I can take a program that technically does work, and then use fundamental programming techniques and best practices to make it more efficient and more useful for the end user. It also shows my understanding of Object-Oriented Programming techniques that will prove valuable to any employer. During my code review, I also recognized other areas for improvement in my original artifact. I performed enhancements regarding user input validation that make the program more reliable. Now if a user selects an option that is not part of the menu, an error message informs the user of their mistake and prompts them to make a valid entry. The original artifact did not have these features in place and would simply crash upon invalid input. These enhancements show that I can use algorithmic principles to solve problems while also making informed design decisions that are appropriate to the specific case.

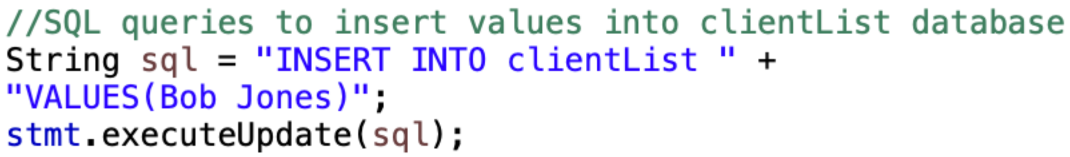
4) **Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals**

    i. I achieved this course outcome in my ClientList artifact by first, converting my existing C++ program into Java. This is proof of my well-rounded programming knowledge in more than one language. It is also proof of my ability to not only learn a language or two, but to understand programming on a more advanced level and be able to implement software solutions to real-world problems. Specifically, in my enhancement regarding user input validation, after translating this program into Java, I implemented a try/catch block that not only validates correct integer input from the menu options, but also handles a mismatch exception that occurred if a user entered a letter instead of a number when making their selection. This error caused the program to crash before my enhancement. Now that error is handled, and the user is prompted with a message telling them to be sure and enter a number from the menu options. Although the program ran perfectly, with valid user input, before this enhancement, it will now continue to run without fail no matter what input it may see from a user. Using this specific method of error and exception handling shows my advanced understanding of available tools in Java and my understanding of computing best practices.



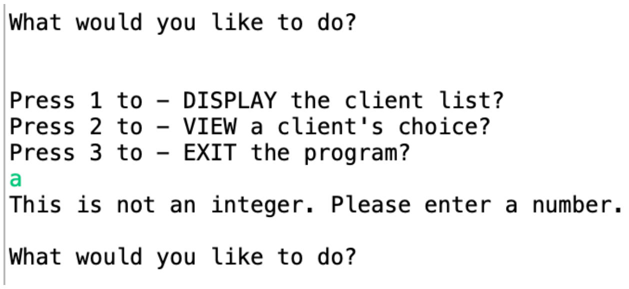
```
//catch handles any user input that is not an integer and prompts user to enter a number
catch(InputMismatchException exception)
{
  //Message to user to obtain valid entry
  System.out.println("This is not an integer. Please enter a number");
  System.out.println();
}
```

ii. I also successfully showcased my proficiency in this course outcome by my enhancement in the Databases category with this artifact. This enhancement meets that criterion by its utilization of trusted database software, MySQL, and taking advantage of its cohesiveness with Java and the Eclipse development environment. By using these combined tools, I have shown my ability to use innovative techniques, skills, and tools to provide valuable computing solutions. First, I chose to use the ArrayList of clients I built for enhancement two to automatically populate a MySQL database. Second, I also created a DatabaseConnection class where I wrote out individual SQL queries that not only portray my experience with SQL databases, but also serves as another way to populate a database from a Java program. The solution offered in this program is an industry-specific one that will meet many relevant business needs: collecting client information, verifying that information, protecting that information, and allowing for expansion in an easy to maintain and simple software solution.



```
//SQL queries to insert values into clientList database
String sql = "INSERT INTO clientList " +
"VALUES(Bob Jones)";
stmt.executeUpdate(sql);
```

5) Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources



```
What would you like to do?


Press 1 to - DISPLAY the client list?
Press 2 to - VIEW a client's choice?
Press 3 to - EXIT the program?
a
This is not an integer. Please enter a number.

What would you like to do?
```

i. I have achieved this outcome by the enhancements I made to my ClientList artifact regarding input validation. Any data input into the system should be validated first. This is a major area for potential vulnerability. This a well-founded Defense in Depth (Johnson, 2020) secure coding standard. After my enhancements that validate all user input before it is allowed into the program, the ClientList artifact is now protected from any malformed or corrupt data entering the system through user input. This is very important for a system that stores client information that is likely sensitive in nature. Having a security mindset means that security is part of the design phase of the program. Security threats are brainstormed much as the UI/UX are. It is no longer enough to handle attacks after they occur. As programmers of today, these attacks must be expected, planned for, and prevented. The only way to come close to preventing attacks in modern computing is to implement security features into the design phase. That way, security threats like buffer overflow, or SQL injection attacks can be prevented. In this simple program, I have showcased my understanding of the importance of protecting customer data, and also that I can implement tools and programming techniques that anticipate security threats and protect user data.

CONCLUSION

This narrative follows my journey through performed enhancements in the categories: Software Design and Engineering, Algorithms and Data Structures, and Databases. I used one artifact, ClientList, for all three categories. This experience has made me a better programmer and has in turn, allowed me to showcase to future employers, professors, as well as myself, that I am ready to become a valuable asset in the software development world of today.

References

Johnson, P. (2020). *Secure Coding: A Practical Guide*. Retrieved 17 June 2022, from https://www.mend.io/resources/blog/secure-coding/.