8-2 Journal: Portfolio Item

CS-340-X6151 Client/Server Development 23EW6

Richard Backscheider

9/4/2023

Southern New Hampshire University

**Prompt**

When writing programs, it is important that you maintain a clean, readable, adaptable, and error- free code, making it easier for other developers to understand and collaborate. When writing the code, a developer should consider using straightforward names for variables, functions, modules, and classes, all of which make the code easier to understand. There are important guidelines that a developer should follow to create a clean, readable, and adaptable code: consistent formatting and naming conventions, keeping lines of code limited in length, use of consistent indentation, adding comments to explain important information, breaking down code into smaller functions or classes, running code frequently to check for errors, and always be consistent with the coding style.

The CRUD class written for this course and its methods can largely be reused in any other scenario—either with the same database or with other databases with a minimum of revisions to conduct successful data queries. The only aspect of the CRUD class used for this course that goes against the convention of being reusable is the fact that login credentials for authentication were hardcoded. That said, it could also easily be revised to prompt the user for authentication information. Additionally, the CRUD class is also customized specifically to the database used in this course because it makes hardcoded selections of the database name and collection within it. Similar to the previous example with authentication, it could be easily adapted to other databases with other schemas. The key advantage of writing a reusable class and methods in general is modularity and efficiency. There is no need to reinvent the wheel every time there is a need for one.

The initial approach is also the most important. If the customer has a written specification sheet, that is a good starting point. However, even with a specification sheet in-hand, it is also best to conduct interviews with not only the customer, but all stakeholders, including end-users. The interviews can help to bridge the information gaps allowing the coder to avoid making assumptions in as to what their intent is. The next step, would be to translate those specifications into a high-level diagram using pseudo code. The pseudo code can be used as a guide to keep the code structure in-line with the customer objectives. Next, if this is a new database, the structure and the schema would need to be determined before the coding begins. If it is an existing database that needs to be revised, the revised structure must be vetted by the stakeholders, to ensure all the necessary information can be stored in the database. At this point a strategy for importing the database data into the new system, should be devised. Finally, adhering to the logic of the pseudo code, coding can actually begin with a focus on modularity and reusability.

A computer scientist is essentially a problem solver. These specialized scientists are trained to translate the solution to real-world problems using code. Computer scientists leverage the speed, accuracy, and efficiency of computers to not only solve problems, but to solve them quickly. Imagine having to sift through thousands of pages of the database in printed form to find the breeds/mixes, sex, and age range for a specified category of rescue dog. It would be a massive waste of time and expense, not to mention paper and manpower in performing a manual query for that information. Now, add the fact that there are two additional categories of rescue dogs that must be found in addition to the first. I think I’ve made my point. By selecting a single pulldown menu item containing a rescue dog category, one could have those results in a matter of seconds.