CSCI-3901-Assignment5-Argument

Name: Pratheep Kumar Manoharan

Banner Id: B00837436

Design principles used:

1. Standardised third party jar is used to establish connection between Java and SQL.
2. User interface layer (MainClass.java), Database layer (db package) and XML content generator (xml package) are separated from each other and loosely coupled from each other.
3. Each class has only one responsibility associated with it and the methods in the same class are highly cohesive (application of single responsibility principle).
4. Each interface has only one responsibility associated with it (application of interface segregation principle).
5. Each class implementation is protected by its interface (application of dependency inversion principle).
6. Replacing an implementation with another implementation is made easy with the interface layer.

Efficient style:

1. Individual Java objects are created for each parent tags like Customer, Address, Category, Product and Supplier.
2. Each child tags inside the parent tag is created as a field inside the object.
3. Adding or removing a tag is simple in this design.
4. The program designed in such a way that the inputs are validated first before proceeding with the execution. In case of invalid inputs, the program will not be executed.
5. DB credentials are not hard coded, and it is added in a separate property class to provide more security.
6. Separate class is used to store all the constants of the program.

Robustness of the program:

1. Input validation, boundary cases, control flow and data flow of the all methods of all classes of the program are tested thoroughly.
2. All the test cases attached are passed with the expected results.
3. Specific exceptions are caught wherever possible and interpreted to users.
4. All exceptions are handled in the best way possible to provide better user experience.
5. Even when database returns null value it is handled before processing.
6. Even when no data is retrieved from the database, the program will write the basic tags of the XML.
7. Even when the program faces unexpected exception in one section of data retrieval it will not affect the other sections. It is handled with individual try and catch block for better robustness.

Perfection of the output:

1. Output is cross verified manually for multiple scenarios. Total number of products and cost in product section is equal to supplier section. Total cost in customer section is also equal to the other sections.
2. XML content is created in the user-friendly readable format.

3. Spacing and line separators are added to improve the readability of the content.
4. Output is created as a string buffer and validated after writing into the output file.

Maintainability of the program (Design elements for minimal future change):

1. In case if we want to add new tag, we just need to make changes in three places.
2. We need to add the tag name as a field in the corresponding object say new field under product tag then a new field in product object and its setter and getter methods.
3. We need to add the column in the corresponding query with the field name as the output column header so that the column value will be set into the object using reflection method.
4. We need to add the getter method name of the new field in the ConstantClass array field in the expected order so that the tag is formed in the correct sequence.
5. Same steps to remove a field from the XML as well.
6. All other details of the program are designed in the generalised way to have less maintainability effort.

Conclusion:

1. The program adheres to all the design solid principles.
2. The data fetched are accurate.
3. The program produces user friendly readable output.
4. The maintainability is easy.

   All the points required for a production code is addressed in this program. So, the code is ready to use as it is.