**cs3307 – Object oriented analysis and design**

**Quiz: OO Concepts, Analysis and Models**

[**Banking System Project**](http://www.cppforschool.com/project/banking-system-project.html)

**Introduction**

In this quiz, you will need to reflect on the program you wrote in Assignment 1. **The following material is particularly relevant for this quiz – please READ it!!!**:

* Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young Ph.D., Jim Conallen, Kelli A. Houston. *Object-Oriented Analysis and Design with Applications* (3rd Edition), Addison-Wesley, 2007. (especially Chapters 1-4)
* Martin Fowler. *UML Distilled*, Third Edition. Addison-Wesley, 2004.
* Other books and material (see course description) that talk about how to design and develop OO systems.

The specific deliverables have been indicated below as **“[D*n* -- <<title>>”]** where **“*n*”** is a number and **<<title>>** is the relevant title for the given deliverable**.**

In your submission**, please first write the precise section number** followed by the deliverable number and title (e.g., (1) (i) [D1 – design]). **Please ensure that deliverables submitted adhere to this specification.**

**Quiz**

1. **Program Design**:
   1. Draw a ***descriptive*** class diagram of the program you wrote in assignment 1. That is, the class diagram should mirror the classes **that are actually in your program**. **[D1 – design]**
   2. ***Classes***: For each class in your diagram (for both interface part and implementation part), please give a reference to where in your program (e.g., file name and line number of your code) that particular class is defined. So, this could result in a simple 2-column table (e.g., class name (interface/implementation part); reference in the program).**[D2 – class reference in code]**
   3. ***Operations***:
   4. Please name the operations (e.g., methods) within each class (in the class diagram). **[D3 – operations within classes]**
   5. Select two classes each with at least two operations (functions/methods) from the table in part **(ii)** above. For each operation, give  the reference for that operation in the program code (i.e., where is that operation in the code?). To make it clearer, you could suitably enhance the table in 1(ii) above to include operation names and their references. **[D4 – operation reference in code]**
2. ***Relationships***:

Explain key relationships across a selected set of classes in your class diagram; **the key points to note are**:

* + 1. Please identify a set of classes that you think has important relationships within it. **[D5 – classes with important relationships]**
    2. Justify the importance you have given to this set of classes over other possible sets of classes in your design. That is, why do you think this set is important – especially from the point of view of the domain (clients, organisation, other stakeholders, etc.)? **[D6 – Justification of importance]**
    3. Explain the relationships (e.g., what relationships exist and why) among the classes in the considered set – especially from the technical point of view of satisfying the domain requirements and importance discussed in (b) above. **[D7 – Relationships: what and why]**

1. **Adequacy of your design resultant from (1) above**: The key issue here is whether your design (in terms of the **elements (i.e., classes, objects, inter-relationships, etc.)** fairly reflects the **problem description (e.g., roles and actions) in Assignment 1**). Please answer the following:
   1. **Show** which aspects in the problem statement in Assignment 1 correspond to what OO elements in your design. This can be shown, for example, in a 2-column table format (Aspect of the problem statement; OO element in design). **[D8 – Correspondence between problem statement and design elements]**
   2. **Assess** (i) above in terms of the adequacy of your design. For example, what aspects of the problem statement in Assignment 1 are **not** directly reflected as OO elements in the design? **[D9 – Assessment of design adequacy]**
   3. **Improve design**: **From the assessment in (ii) above**, answer either (a) or (b) below depending on design adequacy:
      1. If your design is inadequate in certain respects to the problem statement, please redesign your system to make it adequate.
         1. Produce a new design to eliminate the inadequacy. **[D10aA – Improved design]**
         2. Give a revised table (see (i) above) clearly showing the aspects that have been improved. **[D10aB – Table of improved design correspondence]**
      2. If you are claiming that your design is (near) perfect in terms of adequacy then **please justify it**. For example, show how your assessment in (ii) above led you to conclude design adequacy. **[D10b – Justification of (near) perfect design]**
2. With reference to the ***design*** of your program (resultant from improvement made in (2 (iii) (a)(A) or from justification made in 2 (iii) (b)) above), discuss (*max. 0.5 pages for each item unless otherwise indicated*):
   1. “Separation of concerns” ***decisions*** in your design. **[D11 -- Separation of concerns]**
   2. The strength of inter-component vs. that of intra-component relationships in your design. Be specific about the artefacts involved in your design. **[D12 – Strength of inter- vs. intra-component relationships]**
   3. Discuss whether or not the relative balance between inter- and intra-component relationships in your design aligns to certain principles of software design. Again, be specific about design “principles” involved. **[D13 – Principles of software design]**
   4. Discuss why these particular design principles are the way they are. For example, what are the issues underlying these principles? **[D14 – Rationale for principles]**
3. **Structured design** of your program in Assignment 1:
   1. Give an algorithmic (i.e., functional) decomposition of your system programmed in Assignment 1. **[D15 – Functional decomposition]**
   2. Compare and contrast with the object-oriented (OO) design of your system resultant from (2) above. (max. 0.5 pages) **[D16 – Functional vs. OO]**
4. **System dynamics** across design elements:
   1. Describe a typical scenario of system operation (e.g., customer wanting to withdraw some funds from his/her account) in bullet point form. The scenario should include actions from both the user’s side as well as the system’s internal operations. **[D17 – Scenario – user and system aspects]**
   2. Show the execution path (highlight the path for ease of visibility) of this scenario on the functional decomposition diagram (see 4 i), showing what information flows from function to function. **[D18 – Execution path on functional decomposition diagram]**
   3. Generate the execution trace resultant from running the Assignment 1 program using the scenario you have created in (5 i). **[D19 – Execution trace from the program]**
   4. Compare the generated trace against the execution path highlighted (see 5 ii) on the structured design. **[D20 – Comparison of execution traces ]**
5. **Domain and elements of OO design**.
   1. Please list some example categories of objects (e.g., ATM users) in the banking domain. Note that these categories are not limited to the specifics of the problem/project description in Assignment 1; it is meant to be open-ended reflecting on the whole domain of banking. **[D21 – Categories of objects in the banking domain]**
   2. Which of these object types are directly represented in your design (class diagram resultant from (2) above)? **[D22 – Representation of object types]**
   3. Let us assume that your system will be evolving in the coming years due to new requirements.
      1. Tabulate **examples** of some new types of objects that would warrant extension of your class diagram, including inter-relationships between the current class diagram (or parts thereof) with anticipated new classes. Thus, such a table could consist of columns: New objects; Old objects; Inter-relationships. **[D23 – System enhancement -- analysis]**
      2. Show example extensions in 6 (iii) (a) above in a modified class diagram of your system design, highlighting (by colour, for example) the anticipated extensions. **[D24 – System enhancement -- design]**
6. **Models**:
   1. Take the OO design of your system (from (2) above). Using suitable abstractions create a **package diagram** of your system, showing packages and their dependencies. **[D25 – Package diagram]**
   2. List the **criteria** you used, with justification, in creating the package abstractions. **[D26 – Criteria for packaging and justification]**
   3. Briefly, contrast how the high-level design involving packages differs from the lower-level design involving classes. (max. 0.5 pages) **[D27 – High-level design vs. Low-level design]**
   4. Create a **scenario** in bullet-point form that cuts across various packages (e.g., a user function that requires execution of operations from several packages). **[D28 – Scenario cutting across several packages]**
   5. Draw a **sequence diagram** that corresponds to the scenario. **[D29 – Sequence diagram]**
7. **Lessons learnt**:

Please write, as numbered list, ***up to three*** critical lessons learnt on the OO topic in carrying out this quiz. **[D30 – Lessons learnt]**

**Group Work and policy:**

1. This quiz is to be done in the assigned groups.
2. Equitable workload within a group: It is expected that each member of the group will carry his/her weight and contribute equally (within reason). If a member feels that the partner is not carrying his/her weight in the assigned work, please fill out the “**Equitable Workload Statement**” and email it to the Prof. immediately so appropriate actions can be taken. Please not that regardless of the issues within the group, the overall mark for the project will be based on the \*quality\* of the work submitted and not on the \*amount\* of work one did.

**Project requirements:**

1. Please package all the specified deliverables above in **one** document.
2. Header page of the document should contain at least the following:
   * Quiz title.
   * Course number and description
   * Group number AND membership with email addresses.
   * Date
3. Please ensure page numbering.
4. The document should be in PDF format.
5. Please note – **only one submission from each group** will be accepted. We will mark only one and is not clear which one so please do not send more than one!
6. **Submission format:** A pdf file. The file name *must* follow the following format:
   * <”Quiz” ><underscore ”Group” group # underscore><hyphenated last names of group members>
     + E.g., Quiz\_Group 45\_Bloggs-Carpenter
7. **Deadline:** 15th October, 2014 at 11:59:59
8. Recipient email address: cs3307a@gmail.com.

Have fun!