

National Exams May 2014

04-Soft-A3, Software Design

3 hours duration

NOTES:

1. If doubt exists as to the interpretation of any question, the candidate is urged to submit with the answer paper, a clear statement of any assumptions made.
2. This is an OPEN BOOK EXAM.
Any non-communicating calculator is permitted.
3. SIX (6) questions constitute a complete exam paper.
The first five questions as they appear in the answer book will be marked.
4. Each question is of equal value.
5. Most questions require an answer in essay format. Clarity and organization of the answer are important.

04-Soft-A3 Software Design

1.
 - a. What artifacts the software design stage produces? Use an example to explain your the answer.
 - b. Describe the relationship between the stages of software design and requirements engineering in the software development process.
 - c. How artifacts produced from the software design are used in software development process. Use an example to explain your answer.

2.
 - a. Describe the relationship between quality of a software system produced by during software design and the non-functional requirements of the system.
 - b. Among the following software design quality attributes, which quality attributes are system quality attributes which can be evaluated by executing the designed system, and which quality attributes are software quality attributes which cannot be evaluated by executing the designed system, instead by reviewing the designed software.
 - Correctness
 - Reliability
 - Maintainability
 - Performance
 - Portability
 - Security
 - c. Among the software design quality attributes listed above, identify one or more pairs of attributes which may conflict each other. Use examples to explain your answer.

3.
 - a. MVC (Model-View-Controller) is a common software design pattern for GUI based software systems. Explain how the software design principle "separate of concerns" is used in MVC.
 - b. What are typical abstractions in function-oriented software design such as using C programming language? What are typical abstractions in object-oriented software design such as using Java or C++ programming languages?
 - c. In software design, what is information hiding? Why information hiding is important? What software design quality attributes benefit from information hiding?

4.
 - a. Describe the relationships between software architecture and other software development phases: phases, requirements analysis, detailed design, implementation, and testing.
 - b. A software architecture consists of multiple structures each is which are composed of software components, behaviors or properties of components, and relationships or interactions between components. Draw two diagrams of a software system, such as client-server system, with each diagram showing a different structure of the same system.

04-Soft-A3 Software Design

- c. Architecture of any software system always has two views: structural view and behavioral view. What is the difference between these two views? Use an example to explain your answer.
- 5.
- a. What is modularity of a software system? Why modularity is important in software design? What software design quality attributes does modularity support?
 - b. What are modules in function-oriented software systems? What are modules in object-oriented software systems?
 - c. What is decomposition in software design? What is the relationship between software decomposition and quality attributes? Use an example to explain your answer.
- 6.
- a. What is the structure diagram in function-oriented design? Draw a simple 3-level structure diagram.
 - b. In function-oriented design, there are mainly three purposes to decompose software into multiple functions: reuse, abstraction, and modularity. Use an example to explain each of the purposes.
 - c. In function-oriented design, what is the relationship between functions and data structures? Describe two or more ways for multiple functions to share a data structure.
- 7.
- a. Identify classes from the following system requirement description and describe responsibility of each identified class for a walk-in medical clinic.

On each day, before or during the clinic is open, using the system, a patient can sign in to the clinic online, with patient's name and medicare number, at home before going to the clinic. A patient can also view how many patients before him/her currently in the waiting queue (but not other patients' names and medicare numbers). Using the system, when a patient arrives at the clinic, the secretary marks this patient's arrival; if a registered patient missed or complete her/his appointment, the secretary will remove her/his name from the waiting queue. Also the secretary can change the order of a patient in the waiting queue according urgency or some other reasons. Using the system, the secretary can also assign patients to doctors, and a doctor in the clinic can view patients assigned to her/him. Both the doctors and secretary can view the current status of the clinic's waiting queue anytime with full information of registered patients. The doctors and secretary need to be authenticated when accessing the system.
 - b. What are actors of the above system above? What are data entities of the above system above? What classes are needed to link between actors and data entities to support the system's functionality?

04-Soft-A3 Software Design

- c. In object-oriented design, two hierarchies may be designed: class (inheritance) hierarchy and object (aggregation) hierarchy. Use example to explain each hierarchy.
8.
 - a. What are common strategies in interface design to support “easy to learn” usability? Use example to explain your answer.
 - b. What are common strategies in interface design to support “easy to use” usability? Use example to explain your answer.
 - c. What are common strategies in interface design to support “easy to prevent and/or fix user errors” usability? Use an example to explain your answer.
 9.
 - a. What is full signature of a method or function in a module? Use example to explain your answer.
 - b. What is visibility of a method, function, or variable of a module? Explain the following visibility types: public, private, protected.
 - c. What is interface of a module in a function-oriented software system? What is interface of a module in an object-oriented software system? Use examples to explain your answer.
 10.
 - a. Does Is the UML use case diagram a product of software design? What is the relationship between UML use case diagram and software design?
 - b. Draw a UML class diagram for a simple personal address book software application which supports a user to enter, store, retrieve, and delete her personal contact information.
 - c. Draw a UML sequence diagram to show object interactions when a user retrieves a personal contact using the above application.
 11.
 - a. What are design patterns? Where do design patterns come from? What is the relationship between design patterns and reusability?
 - b. Design patterns are specified in a pattern description language which usually consists of the following parts: intent, motivation, applicability, structure, participants, collaborations, consequences, implementation, sample code, known uses, and related patterns. Use one sentence to explain each part.
 - c. The first set of design patterns described in the GoF book, “*Design Patterns: Elements of Reusable Object-Oriented Software*”, includes 23 design patterns in three categories: creational patterns, structural patterns, and behavioral patterns. Explain the differences of the three categories. List the name of one pattern from each category.

Marking Scheme

04-Soft-A3 Software Design

1.
 - a. 3 marks
 - b. 3 marks
 - c. 4 marks

2.
 - a. 3 marks
 - b. 3 marks
 - c. 4 marks

3.
 - a. 4 marks
 - b. 3 marks
 - c. 3 marks

4.
 - a. 3 marks
 - b. 4 marks
 - c. 3 marks

5.
 - a. 3 marks
 - b. 3 marks
 - c. 4 marks

6.
 - a. 3 marks
 - b. 4 marks
 - c. 3 marks

7.
 - a. 4 marks
 - b. 3 marks
 - c. 3 marks

8.
 - a. 3 marks
 - b. 3 marks
 - c. 4 marks

9.
 - a. 3 marks
 - b. 3 marks
 - c. 4 marks

10.
 - a. 3 marks
 - b. 4 marks

04-Soft-A3 Software Design

c. 3 marks

11. Basic concepts of design patterns.

a. 3 marks

b. 3 marks

c. 4 marks