NATIONAL EXAMS - MAY 2019

# 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 a closed book exam with 2 (two) aid sheets allowed written on both sides. No calculator is permitted.

3. SIX (6) questions constitute the exam paper. Answer FIVE (5) of these SIX (6). The first five questions as they appear in the answer book will be marked.

4. Each question is of equal value, and each sub-question in each question is of equal value.

5. Most questions require an answer in essay format. Clarity and organization of the answer are important.

**Question 1**

a. Software design stage in software development process produces two types of artifacts: software architecture design and detailed design. Describe differences and relationships between these two types of artifacts. Use examples to explain your answer.

b. What is basis or input to software design, and how such basis or input affects software design? Use examples to explain your answer.

c. How other stages of software development process use software architecture design and detailed design produced from software design stage? Use examples to explain your answer.

**Question 2**

a. Software design involves two aspects: functionality design and quality design. Describe differences and relationships between the two aspects. Use examples to explain your answer.

b. Among the following two software design quality attributes, which one can be evaluated by executing the designed system, and which one cannot be evaluated by executing the designed system. Explain your answer.
   - Modifiability
   - Performance

c. Among the software design quality attributes listed above, find whether these attributes conflict each other or not. Use examples to explain your answer.

**Question 3**

a. MVC (Model-View-Controller) is a common software design pattern. Draw a diagram to show MVC pattern. Briefly explain how an MVC based system works. What types of software applications can benefit from using MVC pattern in terms of system quality and why?

b. Publish-Subscribe is also a common software design pattern. Draw a diagram to show Publish-Subscribe pattern. Briefly explain how a Publish-Subscribe based system works. What types of software applications can benefit from using Publish-Subscribe pattern inn terms of system quality and why?

c. In designing a software component or module, we usually separate publicly accessible part and private part of the component/module. What are differences and relationships between the two parts? What are benefits of separating these two parts in terms of software quality? Use examples to explain your answer.

**Question 4**

a. A major software design activity is software architecture design. Using an example to explain how architecture design of a software system is based on functional requirements of the system and also based on non-functional requirements of the system.

b. During architecture design activities of a software system, multiple design alternatives may be made based on different considerations of the system's quality requirements. Draw two diagrams to show two alternative architecture designs of a personal address book application which allows a user to add, delete, modify, save, and load contact information. The two designs should have same functionality but with different structures. Explain the differences of the two designs in terms of their quality considerations.

c. In software architecture design, three architectural views need to be considered: module view, component-connecter view, and allocation view. Briefly describe each of the three architectural views and their relationships.

**Question 5**

a. Modifiability is one of the most important software quality attributes. Briefly explain modifiability quality attribute. Briefly explain how modularity in software design can support modifiability quality attribute.

b. Cohesion and coupling are important considerations in designing high-quality software systems. Briefly explain the concepts of cohesion and coupling in software design. What is strong cohesion? Why functional cohesion is strongest? What is loose or weak coupling? Why data coupling is weakest? Why in software design we seek strong cohesion and weak coupling?

c. Decomposition is one of the most import strategies to design high-quality software systems based on systems' quality requirements. A personal address book application can be initially design with a single module. Draw diagrams to show how you can decompose such single module system to a system with multiple modules with separations of user interface, application logic, data management and data storage. Explain your purpose in each decomposition step in terms of making the system have better quality.

**Question 6**

a. In function-oriented design for a software system, we design hierarchies of functions and data structures which are to be used by functions. Using the personal address book application as example, show your design of function hierarchy or hierarchies with function names and their relationships with other functions, and show your design of data structures and their relationships with functions, i.e. which functions use which data structures.

b. In object-oriented design for a software system, we design two types of hierarchies of classes. Class hierarchy shows inheritance relationships between classes. Object hierarchy shows whole-part or aggregation relationships between objects of classes. Using the personal address book application as example, show your design of one or more class hierarchy with class names and their inheritance relationships, and show your design of one or more object hierarchy with class names and their aggregation relationships.

c. Using your answers to a. and b. to discuss pros and cons of the two types of software design methodologies, i.e. function-oriented design and object-oriented design, for this personal address book application, in terms of software development and quality of software products.