

National Examinations - May 2019

17-Comp-B11, Advanced 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. No Calculator permitted. This is a closed book exam with TWO (2) aid sheets allowed written on both sides.
3. You are requested to answer:
 - a. Any five (5) questions in PART I
(only the first five questions of PART I as they appear in your answer book will be marked)
 - b. Any three (3) questions in PART II
(only the first three questions of PART II as they appear in your answer book will be marked)
 - c. Any four (4) questions in PART III
(only the first four questions of PART III as they appear in your answer book will be marked)
 - d. Any two (2) questions in PART IV
(only the first two questions of PART IV as they appear in your answer book will be marked)
 - e. Any five (5) questions in PART V
(only the first five questions of PART V as they appear in your answer book will be marked)
4. All questions have equal weight.

PART I—General principles

Question 1

Software development process models and methodologies used in software development projects influence software design activities. What are differences, in terms of software design activities, between using conventional document-driven process models such as waterfall model and agile development methodologies?

Question 2

User requirements of a software system is the basis for designing the system. What are differences between functional and non-functional requirements of a software system? Briefly describe how design of a software system uses or bases on functional and non-functional requirements of the system.

Question 3

What are software metrics? Why are software metrics important in software design? What is the relationship between software metrics and quality of a software system being designed?

Question 4

Assume that you are asked to develop a personal address book application which supports a user to add, delete, edit, save and load personal contact information. Draw a UML use case diagram to show user requirements of the application. Draw a UML class diagram to show your design of the application. Briefly explain how your design is based on or implements the use cases.

Question 5

In designing a software system, why usually multiple design alternatives or candidate designs, which all can implement the functional requirements of the system, are designed? What is the basis for you to select one design among all design alternatives to be the final best design?

Question 6

In terms of software testing, test cases can be created in different development phases including requirements, design, and implementation phases. What types of test cases can be created in software design phase? Use examples to explain or illustrate your answer.

Question 7

What is software reuse? Why software reuse is important? Briefly describe major activities for software reuse during software design and implementation phases.

PART II—Design by Contract

Question 8

What is design-by-contract in software design? Give a simple example to illustrate your answer. What are pros and cons of using design-by-contract methodology in software design.

Question 9

In design-by-contract based programming, we usually use assertions to guarantee correctness of programs. What is assertion in programming? How assertion is used in design-by-contract based programming? Use an example to illustrate it.

Question 10

In design-by-contract based software design, in designing a software module, we usually specify preconditions and postconditions of the module. Briefly describe how such specified preconditions and postconditions realize design-by-contract in designing a module. Use an example to illustrate your answer. What is the relationship between preconditions/postconditions and assertions?

Question 11

What is class invariant in object-oriented software design? Give an example class design which has class invariant(s). Explain the class invariant(s) of the example class. How class invariant is used in design-by-contract?

Question 12

What is Liskov substitution principle in software design? Give a simple design example that follows the principle. Give a simple design example that violates the principle.

PART III—Patterns

Question 13

Design patterns are usually described by using a pattern language with structured contents as follows: Name, Motivation, Applicability, Forces, Solution Structure, Consequences, Implementation, Sample code, known applications, Relation to other patterns. Use one to three sentences to explain each of the above topics.

Question 14

The “Gang of four” design patterns are classified as class inheritance-based patterns or object-composition based patterns. What do these two terms mean? Give an example design pattern of each kind and justify its classification.

Question 15

Use examples to describe the differences between the Factory Method design pattern and the Template Method design pattern.

Question 16

Use examples to describe the differences between the State design pattern and the Strategy design pattern.

Question 17

Assume that you are asked to develop a personal address book application which allows a user to add, delete, edit, save, and load personal contact information. You are also asked to use the Façade design pattern to design the application. Draw a diagram to show your design of the application and explain where and how the Façade design pattern is implemented in the design.

PART IV—GUI Design

Question 18

Assume that you are asked to develop a personal address book application which allows a user to add, delete, edit, save, and load personal contract information. Draw a UML class diagram to show how you can use the Model-View-Controller (MVC) architectural pattern to implement the application.

For each design goal below, indicate whether the MVC architecture in your design may help or hurt. Justify your answer in each case.

- extensibility of the system
- response time
- modifiability of the design

Question 19

In object-oriented software design, what are boundary classes? What are control classes? What are entity classes? How is the “separations of concerns” software engineering principle applied in such classification of design classes? Among the three types of classes, which type of classes involves GUI design? Using your design for the personal address book application to illustrate these three types of classes.

Question 20

Use examples to explain how using the Command design pattern and Memento design pattern can help design high quality GUI for software applications.

Question 21

Software usability has three goals for software applications: easy to learn, easy to use, and help prevent and/or fix user errors. Assume that you are asked to develop a personal address book application which allows a user to add, delete, edit, save, and load personal contract information. Describe your design strategy for achieving each of the above usability goals for the system.

PART V—C++/Java and Modular Programming

Question 22

One of the important object-oriented programming principles is “Favor object composition, over class inheritance.” Use C++ or Java code examples to explain this principle.

Question 23

In designing a class in C++ and Java, we can classify and declare variables and methods in the class as private, protected, and public. What is the meaning of such classification? Why such classification is important to support encapsulation and modularity software engineering principles?

Question 24

Give sample code in C++ or Java to illustrate the following terms in object-oriented software design and programming:

- polymorphism
- dynamic binding
- overloading
- overriding.

Question 25

Using example C++ code to show how a C++ class can allow other classes access its private variables and methods, which is not supported or allowed in Java. What are advantages and disadvantages of the C++ approach? In your opinion, why Java does not allow it.

Question 26

Another important object-oriented programming principle is “Programming to interface, not to implementation”. Use simple example code in C++ or Java to explain this principle.

Question 27

Java supports “interface” in addition to “class”, but C++ has only “class”. What is the difference between interface and class? How interfaces in Java can be simulated or implemented in C++ using classes?

Question 28

In modular programming, a software system is partitioned into multiple software modules can those modules can be related by dependency, association or coupling relationships between them. A software module is usually designed with public part and private part, where the public part can be accessed by other modules, but the private part cannot be accessed by other modules. Briefly describe how modular programming can be implemented in programming using C++ and Java. Use examples to illustrate your answer.