四川大学期末考试试题（闭卷）
（2023——2024 学年第 1 学期） B 卷

课程号: 课序号: 课程名称: 任课教师: 成绩:
适用专业年级: 学生人数: 印题份数: 学号: 姓名:

考生承诺

我已认真阅读并知晓《四川大学考场规则》和《四川大学本科学生考试违纪作弊处分规定（修订）》，郑重承诺:

1、已按要求将考试禁止携带的文具用品或与考试有关的物品放置在指定地点;

2、不带手机进入考场;

3、考试期间遵守以上两项规定，若有违规行为，同意按照有关条款接受处理。

考生签名:

一、单项选择题（本大题共 20 小题，每小题 1 分，共 20 分）

1. Software includes instructions, data structures and (   ).
   a. computer programs   b. descriptive information
   c. a series of charts   d. list of backlogs

2. In the deployment activity, the software is (   ) to the customer and provides feedback.
   a. debugged   b. delayed   c. delivered   d. developed

3. The (   ) process flow executes each of the five framework activities in sequence, beginning with communication and culminating with deployment.
   a. evolutionary   b. parallel   c. iterative   d. linear

4. The (   ) method is one of the popular agile methods.
   a. XP   b. TP   c. CP   d. AP

5. In Scrum process flow, development proceeds by breaking the project into a series of incremental prototype development periods 2 to 4 weeks in length called (   ).
   a. sticks   b. strings   c. springs   d. sprints

6. Scenario-based models depict software requirements from the (   )'s point of view.
   a. developer   b. leader   c. user   d. viewer

7. Activity diagrams describe (   ) behavior of the software.
   a. dynamic   b. excited   c. static   d. whole

8. A nonfunctional requirement can be described as a performance attribute, a security attribute, or a general (   ) on a system.

a. collection   b. constraint   c. configuration   d. review

9. Actors in a use case are the different people or (   ) that use the system within the context of the function and behavior that is to be described.

a. events   b. devices   c. states   d. transitions

10.Which of the following should not be considered as candidate analysis class? (   )

a. events   b. external entity   c. operations   d. structure

11. To define attributes of analysis classes, (   ) should be extracted from use-cases.

a. nouns and noun phrases   b. adverbs and adverb phrases

c. verbs and verb phrases   d. adjectives and adjective phrases

12. In the context of object-oriented modeling a component contains (   )

a. attributes and operations   b. instances of each class

c. roles for each actor   d. a set of collaborating classes

13. (   ) is achieved by developing modules with "single-minded" function and an "aversion" to excessive interaction with other modules.

a. Design class   b. Functional independence

c. Separation of concerns   d. Stepwise refinement

14. The Interface Segregation Principle means "One general purpose interface is (   ) than many client-specific interfaces".

a. better   b. equal   c. same   d. worse

15. (   ) is not area of concern in the design model.

a. Scope   b. Deployment   c. Data   d. Architecture

16. The (   ) design describes how the software communicates with systems that interoperate with it, and with humans who use it.

a. component   b. data   c. interface   d. deployment

17. (   ) testing is usually performed early in the testing process.

a. White-box   b. System   c. Integration   d. Black-box

18. Bottom-up integration testing has as it's major advantage that no (   ) need to be written.

a. driver   b. cluster   c. stub   d. module

19. (   ) testing leads to the construction and test of the complete software architecture.

a. Validation   b. Integration   c. System   d. Unit

20. Abstraction and (   ) are complementary concepts.

a. responsibilities   b. review   c. refactor   d. refinement

二、 判断分析题（本大题共 2 小题，每小题 5 分，共 10 分）

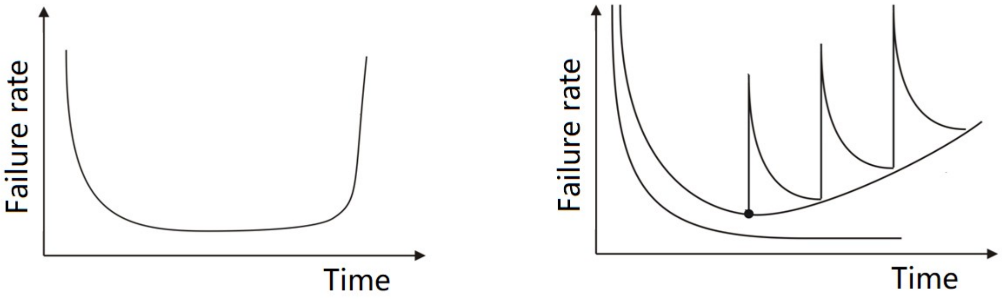
1. "Computer software needs to evolve over time." Do you think this statement is true? Why?

2. "Components should try to exhibit as low cohesion as possible." Do you think this statement

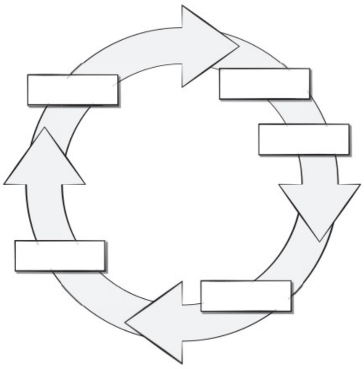
is true? Why?

三、看图分析题（本大题共 2 小题，每小题 5 分，共 10 分）

1. The following figures depict failure rate as a function of time for hardware and software. Explain the meaning of the two figures and describe the difference between hardware and software.



2. According to the following figure, describe the characteristics of the prototyping model, as well as the advantages and disadvantages of the model when applied to development.



四、问答题（本大题共 4 小题，每小题 5 分，共 20 分）

1. What should be done at project inception to establish the groundwork?
2. Modularity is a basic design concept. How to decompose a software solution to obtain the best set of modules?
3. Should testing be performed by developers or an independent test group? Why?
4. Describe agility in your own words.

五、综合题（本大题共 4 小题，每小题 10 分，共 40 分）

1. Consider the following use case for version 1.0 of the SecurityHouse system, develop a state diagram for the ControlPanel object.

“The homeowner uses the control panel to key in a six-digit password. The password is compared with the valid password stored in the system. If the password is correct, the control panel awaits to select further action. After the homeowner activates the sensor, the control panel returns to its initial state. If the password is incorrect, the control panel reset itself for additional input, and the homeowner will need to re-enter the password; if the password is entered incorrectly more than three times, the control panel is locked; when the lockout time exceeds one minute, the control panel returns to its initial state.”

2. Based on the use case in 1, develop a sequence diagram.

3. In version 1.0 of the SecurityHouse system, the homeowner operates it using the keys on the control panel. In an advanced version of the system, control panel functions may be implemented via a mobile platform (e.g., smartphone or tablet). Develop a corresponding class diagram that follows the open-closed principle.

4. Condider the following pseudocode:

```
i = 1;
total.input=total.valid = 0;
sum=0;
 DO WHILE (total.input < 100)
    increment total.input by 1;
    IF value[i] >= minimum AND value[i] <= maximum
       THEN increment total.valid by 1;
              sum= sum + value[i];
       ELSE skip
    ENDIF;
    increment i by 1;
 ENDDO
```

(1) Draw the flow graph with simple conditions.

(2) Compute cyclomatic complexity.

(3) To test the functionality of the pseudocode using boundary value analysis, how should the test cases be designed? Provide test cases.

一、单项选择题（本大题共 20 小题，每小题 1 分，共 20 分）

b c d a d c a b b c a d b d a c a c b d

二、 判断分析题（本大题共 2 小题，每小题 5 分，共 10 分）

1. Answer：true

Software must be adapted to meet the needs of new computing environments or technology.
Software must be enhanced to implement new business requirements.
Software must be extended to make it interoperable with other more modern systems or databases.
Software must be re-architected to make it viable within a network environment.

2. Answer：false

Hard to implement, test, and maintain. Based on this core point it is then elaborated.

三、看图分析题（本大题共 2 小题，每小题 5 分，共 10 分）

1.

(2’) Hardware exhibits relatively high failure rates early in its life; defects are corrected, and the failure rate drops to a steady-state level for some period of time. As time passes, however, the failure rate rises again.
(2’) The failure rate curve for software should take the form of the “idealized curve”. Undiscovered defects will cause high failure rates early in the life of a program. However, these are corrected and the curve flattens as shown. The idealized curve is a gross oversimplification of actual failure models for software.
(1’) Software doesn’t wear out. But it does deteriorate.

2.

(1’) The prototyping paradigm begins with communication……deployment.
(2’) Pros: There is a reduced impact of requirement changes. The customer is involved early and often. It works well for small projects. There is reduced likelihood of product rejection. (Answer two of these points)
(2’) Cons: Customer involvement may cause delays. There may be a temptation to “ship” a prototype. Work is lost in a throwaway prototype. It is hard to plan and manage. (Answer two of these points)

四、问答题（本大题共 5 小题，每小题 6 分，共 30 分）

1.

Identifying Stakeholders; Asking Questions; Recognizing Multiple Viewpoints; Working Toward Collaboration;

2.

(3’) 从模块大小与开发成本的角度分析；

(2’) 从信息隐藏的角度分析。

3.

(1’) Both

(2’) The software developer is always responsible for testing the individual units (components) of the program, ensuring that each performs the function or exhibits the behavior for which it was designed.

(2’)The role of an independent test group (ITG) is to remove the inherent problems associated with letting the builder test the thing that has been built. Independent testing removes the conflict of interest that may otherwise be present. After all, ITG personnel are paid to find errors.

4.

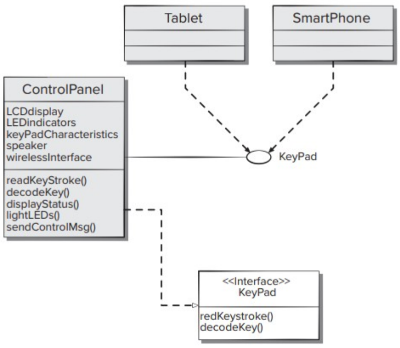
(1’) The pervasiveness of change is the primary driver for agility.

(4’) Furthermore, it also encompasses the philosophy espoused in the manifesto noted at the beginning of this chapter. It encourages team structures and attitudes that make communication (among team members, between technologists and business people, and between software engineers and their managers) more facile. It emphasizes rapid delivery of operational software and deemphasizes the importance of intermediate work products (not always a good thing); it adopts the customer as a part of the development team and works to eliminate the “us and them” attitude that continues to pervade many software projects; it recognizes that planning in an uncertain world has its limits and that a project plan must be flexible.

五、综合题（本大题共 4 小题，每小题 10 分，共 40 分）

1. Open question, open answer

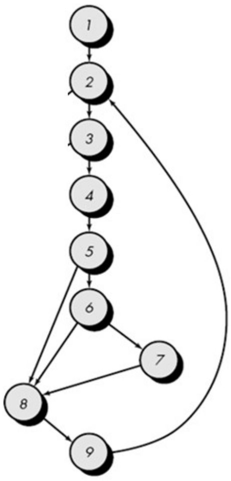
2. Open question, open answer



3.

4.

(1)



(2) V(G)=4　(3) open answer