COMP 3111: Software Engineering

Review 2

Q1. Defensive programming means that you

1. do not let anyone else read your code.
2. do not let input data crash your program.
3. do not let anyone else change your code.
4. do not use object-oriented programming languages.
5. do not let anyone else test your code.

Q2. The primary purpose of refactoring code is to

1. reduce the number of input parameters.
2. isolate bugs by rewriting the code.
3. add assertions to the code.
4. improve the internal structure of the code.
5. improve the interaction among components.

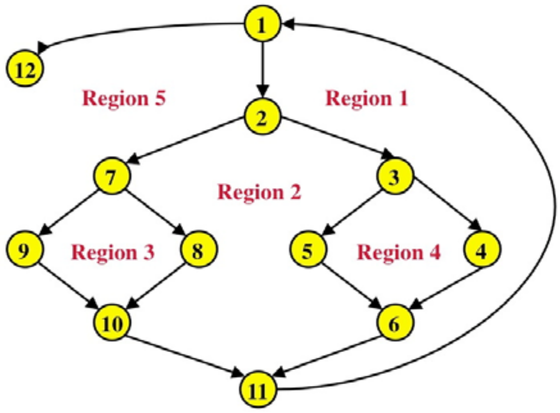
Q3. Basis path testing ensures that we have tested at least

1. all interactions between programs.
2. all the ways of executing the loops in a program.
3. all the statements in a program.
4. all the data structures in a program.
5. all the inputs to a program.

Q4. What is the cyclomatic complexity of the procedure in the figure?

```
Procedure: process records
  Do while records remain
    read record
    If record field 1 = 0
      If record field 2 = 0
        store in buffer
        increment counter
      Else
        print record
      Endif
    Else
      If record field 3 = 0
```

```
12.             store in file
13.             increment counter
14.           Else
15.             delete record
16.           Endif
17.         Endif
18.     Enddo
    End
```



Flow graph node to procedure statement mapping:

Node	Statement
1.	1
2.	2, 3
3.	4
4.	5, 6
5.	7, 8
6.	9
7.	10, 11
8.	12, 13
9.	14, 15
10.	16
11.	17
12.	18

1. 3
2. 4
3. 5
4. 6
5. 7

Q5. When testing a *nested* loop, we initially test the inner loop while holding the outer loop at

1. its minimum value.
2. its middle value.
3. its maximum value.
4. both its minimum and maximum value.
5. any value.

Q6. Black box testing uses test values at the boundaries of a subdomain because

1. these values are easier for us to figure out.
2. this will make integrating components easier.
3. errors are more likely to occur here.
4. black box testing only works for such values.
5. these values can be given to us by the users.

Q7. One of the required inputs for a program that does room scheduling for a public venue is the day of the week, which is input as Sunday, Monday, etc. When testing this program for the input day of the week, what are the **minimum number of test values** that you would use?

1. 5
2. 7
3. 8
4. 9
5. 10

Q8. Black box testing techniques are used in what type of testing?

1. unit
2. condition
3. loop
4. integration
5. data flow

Q9. Test cases for state-based testing can be derived from the

1. use case model.
2. state machine diagrams.
3. basis paths.
4. domain model.
5. nonfunctional requirements.

Q10. The purpose of an acceptance test plan is to

1. verify the acceptability of the code structure.
2. specify the criteria for determining whether the system is finished.
3. list all the requirements for the system.
4. define the scope of the system development.
5. define the goals of the system.

Q11. One purpose of system analysis and design is to

1. determine the cost of developing the system.
2. determine the time required to implement the system.
3. capture the system's nonfunctional requirements.
4. adapt the requirements to the implementation environment.
5. obtain the client's approval for developing the system.

Q12. In the Model-View-Controller (MVC) architectural pattern, the Model represents

1. the objects used to render data in the user interface.
2. the process control mechanism used by the system.
3. the controls with which the user interacts.
4. [✓ the data viewed and manipulated by the user.]
5. the business logic of the application.

Q13. System analysis and design distributes the functionality of a use case into boundary, entity and control classes because

1. these are the only kinds of classes available to us.
2. these classes are easier to test.
3. [✓ we want to isolate specific types of changes to specific types of classes.]
4. state machine diagrams use these classes.
5. object interaction is more easily described using these classes.

Q14. According to the best practices of analysis object interaction, which one of the following interactions **should not** be allowed?

1. interactions between a boundary object and a control object
2. interactions between two control objects
3. interactions between a boundary object and an entity object
4. [✓ interactions between two boundary objects]
5. interactions between a control object and an entity object

Q15. A design class is most cohesive when it

1. is designed by only one person.
2. does not require database access.
3. has few dependencies to other classes.
4. is used in only one scenario of a use case.
5. [✓ has responsibilities that are closely related.]

Q16. A state machine diagram describes the behaviour of

1. [✓ an object.]
2. a use case.
3. an operation.
4. an actor.
5. a class.

Q17. A state machine diagram responds to every event that

1. occurs.
2. occurs provided it is not processing an activity.
3. changes a value of the object's attributes.
4. [✓ triggers a transition.]

5. sends a message.

Q18. Which of the following **is not true** about design patterns?

1. They help novices behave like experts.
2. They are described using program code.
3. They represent a solution to a problem in a context.
4. They can be used to handle nonfunctional requirements.
5. They make extensive use of inheritance and delegation.

Q19. The **Strategy** design pattern is used to

1. defer instantiation of a class to its subclasses.
2. provide a placeholder for another object.
3. decouple an abstraction from its implementation.
4. restrict a class to have only one instance.
5. encapsulate an interchangeable family of algorithms.

Q20. The **Mediator** design pattern is used to

1. provide a unified interface to a subsystem.
2. provide a placeholder for another object.
3. decouple an abstraction from its implementation.
4. restrict a class to have only one instance.
5. encapsulate how a set of objects interact.

Q21. If an organization has a standards handbook for software development, then which of the following probably **is not true**?

1. A project can decide to not follow any standards.
2. A project can decide to ignore a standard.
3. A project can decide to use a standard as is.
4. A project can decide to modify a standard.
5. A project can decide to create a new standard.

Q22. The cyclomatic complexity of a program can tell us something about which of the following external attributes (design goals) of software quality?

1. usability - how easy it is to use
2. learnability - how easy it is to learn how to use
3. safety - how likely it is to be error free
4. integrity - how well it prevents unauthorized access
5. installability - how easy it is to install

Q23. In terms of software quality assurance, the purpose of both the SEI Process Capability Maturity Model (SEI-CMM) and the People Capability Maturity Model (PCMM) is to:

1. educate and train managers.
2. provide feedback on current practices.
3. assess and improve current practices.
4. develop best practices.
5. improve the standards handbook.

Q24. Which of the following **cannot be shown** on a Gantt chart?

1. the task dependencies
2. the time estimates for each task
3. the resource assignment for each task
4. the tasks that lie on the critical path
5. the progress of each task

香港科技大學

The Hong Kong
University of
Science & Technology