

Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page. The exam has 10 questions with equal weight. Some questions have multiple parts. Point totals are given in the margin.

Name: _____ **Answer Key** _____

1. _____ **10**
2. _____ **10**
3. _____ **10**
4. _____ **10**
5. _____ **10**
6. _____ **10**
7. _____ **10**
8. _____ **10**
9. _____ **10**
10. _____ **10**
- Total _____ **100**

No notes or books may be used on the exam. If you have any questions, please raise your hand and I will try to answer them.

1. Answer the following general questions related to software engineering:

2

(a) According to Brooks, what quality distinguishes a programming product from a program?

Answer: A program becomes a programming product if it provides general functionality and has been tested and documented thoroughly.

2

(b) How are abstraction and refinement related?

Answer: Refinement is the gradual elaboration of details of a system in a top-down manner, but stops at a certain level where abstraction is then used to hide the lower-level details.

2

(c) What is the importance of building a system with low coupling?

Answer: If a system has low coupling, a change made to or error occurring in one part of a system is less likely to affect other parts of the system.

2

(d) What is the semantic gap?

Answer: The semantic gap refers to the intellectual distance (i.e., difference in levels of abstraction) between stages in a software process.

2

(e) What is scaffolding?

Answer: Scaffolding is infrastructure placed in code when it is written to facilitate testing and debugging.

2. The waterfall model is described as a “document-driven” model, whereas the spiral model is described as a “risk-driven” model. Answer the following questions regarding these models:

2

(a) What determines when to progress to the next phase of the waterfall model?

Answer: We progress to the next stage when the document (requirements document, specification, design document, etc.) is completed and approved.

2

(b) What determines when to progress to the next iteration of the spiral model?

Answer: We progress to the next iteration using the results of risk analysis on the previous iteration.

2

(c) How are changes in requirements accommodated in the waterfall model?

Answer: A change in requirements is accommodated by going back to the previous stage or stages and regenerating the document with the appropriate change and then performing the stage or stages again.

2

(d) How are changes in requirements accommodated in the spiral model?

Answer: A change in requirements is accommodated as the result of risk analysis in the next iteration.

2

(e) Which model has higher process visibility (i.e., the ability to discern how close the project is to completion)? Briefly explain your answer.

Answer: The waterfall model has higher process visibility since it progress linearly from start to completion.

3. Define the components, connectors, and constraints of the following software architectures:

2

(a) object-oriented

Answer:

- components: objects
- connectors: messages
- constraints: must know the identify of an object in order to communicate with it

2

(b) layered

Answer:

- components: layers of modules or virtual machines
- connectors: function or method calls between layers
- constraints: a layer can only communicate with the layer directly below (or above) it

2

(c) pipe and filter

Answer:

- components: filters
- connectors: pipes
- constraints: pipes are unidirectional queues; filters are independent

2

(d) repository

Answer:

- components: data store and clients
- connectors: reads and write to the data store
- constraints: clients communicate only through the data store

2

(e) event-based

Answer:

- components: objects or modules
- connectors: events and traditional function calls
- constraints: modules must know the names of events in order to register with the event manager

4. Answer the following questions related to software testing:

2

(a) What is the goal of software testing?

Answer: The goal of testing is find errors in software.

2

(b) What makes a test case truly successful?

Answer: A test case is truly successful only if it finds a previously undiscovered error.

2

(c) Why is exhaustive testing not possible?

Answer: Exhaustive testing is not possible because the number of possible inputs for any nontrivial program is too large to try all of them.

2

(d) Why is it best to write test cases before writing the code?

Answer: By writing the test cases before you write the code, you are testing the requirements and design and are not biased by the implementation.

2

(e) What are (in order) the six stages of testing?

Answer:

1. unit testing
2. integration testing
3. function testing
4. performance testing
5. acceptance testing
6. installation testing

5. Answer the following questions regarding unit testing:

2

(a) What is black box testing?

Answer: Black-box testing is testing conducted at the software interface to demonstrate that correct inputs yield correct outputs.

2

(b) What is white box testing?

Answer: White-box testing is testing conducted on the individual program components to demonstrate that intermediate results on each path through the code are correct.

2

(c) What are two possible ways to unit test the code without executing it?

Answer:

- formal proofs
- symbolic execution
- code reviews

2

(d) What are three possible characteristics of object-oriented code that make it easier to test?

Answer:

- reuse: reused classes should already be tested
- early interface definition: test planning is easier
- small methods: easier to test a smaller piece of code
- encapsulation: private data should be protected
- modularity: less chance of errors rippling through the system

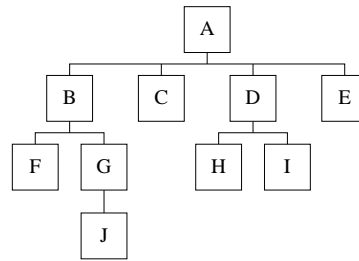
2

(e) A program expects an alphanumeric login name with a maximum of eight characters as input. What are four possible good test inputs?

Answer:

- a string with non-alphanumeric characters
- a string with more than eight characters
- a string with exactly eight characters
- an empty string

6. Consider a system with the following modular structure:



Using a big-bang integration strategy, the following modules would be tested at each step:

1. A, B, C, D, E, F, G, H, I, J
2. (A..J)

Show the steps and modules tested using the following integration strategies:

5

(a) bottom-up

Answer:

1. J
2. F, (G, J), H, I
3. (B, F, G, J), C, (D, H, I), E
4. (A..J)

5

(b) top-down

Answer:

1. A
2. (A..E)
3. (A..I)
4. (A..J)

7. Answer the following questions related to software maintenance:

2 (a) What are the four different types of maintenance?

Answer:

- corrective
- perfective
- preventive
- adaptive

2 (b) What are the four different types of software rejuvenation?

Answer:

- restructuring
- redocumentation
- reverse engineering
- re-engineering

2 (c) What are two possible ways that you can plan for and reduce maintenance costs?

Answer:

- use an evolutionary process model that accommodates change
- design systems with high coupling and low cohesion
- perform preventive maintenance

2 (d) What is the relationship between re-engineering and software quality?

Answer: The goal of re-engineering is to improve the quality of the software.

2 (e) Why are automated analysis tools needed to help understand programs?

Answer: Automated tools are necessary because an implementation is too large to be understood by hand and high-level documentation for a system typically does not exist.

8. Answer the following questions related to software quality:

2 (a) What are three possible qualities that affect product operation?

Answer:

- correctness
- reliability
- usability
- integrity
- efficiency

2 (b) What are two possible qualities that affect product revision?

Answer:

- maintainability
- flexibility
- testability

2 (c) What are two possible qualities that affect product transition?

Answer:

- portability
- reusability
- interoperability

2 (d) What is the relationship between quality and software testing?

Answer: Passing the test suites does not guarantee quality software. However, failing a test certainly guarantees that the software is not of high quality.

2 (e) What is the difference between an extendable product and an extensible product?

Answer: An extendable product can be built upon or extended. An extensible product is one that can be used for purposes other than the intended.

9. Briefly justify whether each of the following is used for validation, verification, or both:

2

(a) acceptance testing

Answer: Acceptance testing is used for validation since the tests are written by the customer based on the requirements.

2

(b) white box testing

Answer: White box testing is used for verification since it is done at the level of the implementation to ensure that the code performs correctly.

2

(c) specification review

Answer: A specification review is used for validation since a specification is simply a formalization of the requirements.

2

(d) black box testing

Answer: Black box testing is used for validation since the tests are done at the interface level and can therefore test the requirements. It can also be used for verification in unit testing to check that the implementation satisfies the design.

2

(e) code review

Answer: A code review is used for verification since it checks the implementation against the design.

- 5 10. (a) Would a customer prefer a product with minimal, flawless functionality or a product with full, but flawed, functionality? Explain your answer.

Answer: A customer would probably prefer a product with minimal, flawless functionality. A product, even a minimal one, that correctly satisfies the requirements is usually more important than a product with many features that work correctly only part of the time. Releasing a flawed product, even one with lots of functionality, damages the reputation of the company and the product. On the other hand, releasing a flawless product still allows your customer to gain a foothold in the market and attract users who will likely stay with the product and anticipate the next product release.

A customer may prefer a product with full, but flawed, functionality if full functionality were required to ensure a successful product in the market. But, these situations are usually rare and often involve products for critical or real-time systems, where partial functionality is not possible. In these cases, however, flaws cannot be tolerated.

- 5 (b) Approximately 60% of all software effort and cost is spent performing software maintenance. This percentage has not changed significantly over the last 20 years. Why do you think this is? Are we really developing such horribly flawed products? What do these facts and your answer say about the way in which we write or should write software?

Answer: Although 60% of all effort is spent on maintenance, maintenance is more than just fixing bugs. Approximately half of this effort is spent on perfective maintenance (i.e., enhancement), while only a small part is spent on corrective maintenance (i.e., fixing bugs). Rather, we seem to be developing products that are successful, and therefore need to be enhanced or adapted to new platforms. However, we would like this cost to be lower and therefore we need to spend more time developing software that is easier to expand and enhance and use methodologies that plan for future modifications to requirements.