CS 30700 Midterm Exam, Fall, 2015

Chapter 1 -- Software and Software Engineering

(1) For large software systems why are software engineering development techniques needed?
a. Sometimes the solution is to buy, not assemble or build
b. An engineering process involves applying well-understood techniques in an organized and disciplined way
---c. Large systems cannot be completely understood and developed by one person
d. There is no software that can be reused
e. There must be a lot of documentation for a project

(2) What is the Software Quality attribute concerned with system failure and its associated consequences?
a. Usability
b. Performance
---c. Availability
d. Reliability
e. Maintainability

(3) What Software Quality attribute is most related to long term quality?
a. Usability
---b. Scalability
c. Performance
d. Security
e. Testability

(4) Which best describes a Brownfield Project?
a. Project that lacks any constraints imposed by other systems
b. "From scratch" project, new development
---c. A software system that must work with existing software system(s)
d. Reengineering project
e. Corrective project

SCRUM Software Design Framework

(5) Which of the following does NOT describe a Sprint?
a. Complete, useable, and potentially-releasable product increment is created
b. Development Team composition remains constant
c. Software Quality attributes remain constant
---d. Duration of 2 weeks to 6 months
e. A new Sprint starts immediately after the conclusion of the previous Sprint

(6) All the following are advantages of the Daily Scrum (or Stand-up Meeting) except which one?
a. Improves communication among team members
---b. Sets priorities for the Weekly Scrum Meeting and Monthly Scrum Meeting
c. Identifies and removes impediments to development
d. Highlights and promotes quick decision-making
e. Improves the Development Team's level of project knowledge

(7) Which of the following is a demo of the product being produced?
---a. Sprint Review Meeting
b. Sprint Retrospective Meeting
c. Daily Scrum
d. Sprint Planning Meeting
e. Stand-up Meeting

Software Engineering: Working As A Team

(8) What is the so-called "herd" mentality of large teams?

a. Meetings are generally shorter when fewer people need to speak
b. Large teams make it easier to develop a common purpose with mutual goals and mutual accountability
c. The larger the team of people, the easier it is for the team to work well together
d. People tend to break away from the herd and state their own opinions
---e. People tend to go along with popular opinion rather than thinking for themselves

(9) A phase of team development is the Forming period. Why is this often called the "honeymoon" period?

a. Because teams can get stuck in this phase and as a result can fail
b. Because during this phase team members begin challenging and disagreeing with one another
---c. Because this phase occurs when a team first comes together and everyone is nice to each other
d. Because in this phase team members start offering ideas and suggestions
e. Because in this phase team members work hard together

(10) Which of the following illustrates "lack of follow through"?

a. A team member offers excuses about why he did not get his tasks done
b. A team member does not do what he promises he will do
c. A team member checks out code from the repository, but never gets around to modifying it and committing it
d. A team member complains that no one ever helped her with some GUI issues and that is why she did not finish the design
---e. All of the above.

(11) When there is a difference of opinion on a team, what is the responsibility of each team member?

---a. To speak frankly and openly, to risk conflict with other team members
b. To become entrenched in your position
c. To refuse to negotiate with other team members
d. To communicate mostly with team members who agree with you
e. To convince the others that your position is the only good one

(12) What is the best way to make decisions on a Software Engineering team?

a. decision made by Project Owner
b. decision made by Project Coordinator
c. decision made by Team Leader
d. majority rule
---e. consensus

(13) Which best characterizes a Software Engineering Team Leader?

a. Use negative reinforcement to force members to do what you want them to do
b. He does less work than other team members because of his Team Leader position
---c. She sets an example by volunteering to do jobs that no one else wants to do
d. Never ask the team to change strategies
e. Represent the team to the Project Coordinator, but talk very little to team members

Version Control

(14) Version Control allows us to do all the following except which one?
a. Keep everything of importance in one place
b. Manage changes made by the team
c. Track changes in the code and other items
---d. Ensure that all committed code matches the chosen software architecture
e. Avoid conflicting changes

(15) You can introduce a completely new feature or concept and not mess up the working code by ...?
a. Change/Bug Tracking
b. Committing
---c. Branching
d. Reverting
e. Merging

Chapter 4 -- Developing Requirements

(16) Which of the following is NOT a Functional Requirement?
a. What computations the system should perform
---b. What platform (hardware) the system should be run on
c. What inputs the system should accept
d. What outputs the system should produce
e. What data the system should store that other systems might use

(17) A use case should...
---a. Describe the user's interaction with the system
b. Cover the first few steps of a task from the beginning
c. Be mapped to a particular user interface design
d. Include actions a user does manually
e. Include internal actions of the system

(18) What do we call a Use Case that represents several similar use cases?
a. Specialization
b. Extension
c. Inclusion
---d. Generalization
e. Superclass

(19) How does a User Story differ from a Use Case?
---a. A User Story does not contain as much detail as a related Use Case
b. Not all Use Cases can be turned into User Stories
c. Use Cases are fixed early in the project, but User Stories may be modified throughout development
d. You may create as many Use Cases as you like, but you may only create as many User Stories as you can finish during development
e. A User Story is a set of Actor Actions and System Responses, while a Use Case is a statement like "As a ___, I would like to ___"

Chapter 3 -- Basing Software Development on Reusable Technology

(20) All the following are problems impeding reuse and reusability except which one?
a. Why take the extra time needed to develop something that will benefit other projects?
b. Management may only reward the efforts of people who create the final products
c. Software is often created in a hurry and without enough attention to quality or reuse
---d. Employing reusable components can often complicate design
e. It takes more time to develop software that is reusable

(21) In a Framework, what do we call classes or methods used by the Framework that have to be supplied by the user of the Framework?
a. APIs
b. hooks
---c. slots
d. over-ridden classes or methods
e. virtual classes or methods

(22) All the following are true about the Client-Server Architecture except which one?
a. Client-Server is an example of a Distributed System
b. Each server can run on different hardware platforms
c. Each client can run on different hardware platforms
d. Each server can provide services to several clients
---e. Each client can access only one server

(23) All the following are advantages of the Client-Server Architecture except which one?
a. Work can be distributed among different machines
---b. Clients and servers cannot be designed separately
c. Clients can access a server's functionality from a distance
d. Clients and servers can be simpler
e. All data is distributed among many multiple servers

Chapter 5 -- Modelling With Classes

(24) In a UML diagram if the relationship between class A and class B is 1 to many (1 to *), how many B objects can there be for each A?
---a. any number including zero of them
b. any number of them, with a minimum of one
c. any number of them, but must be more than one
d. exactly one of them
e. zero of them

(25) Sometimes, an attribute that concerns two associated classes Alpha and Beta cannot be placed in either of the classes. It is put in an Association Class Gamma. What are the relationships among Alpha, Beta, and Gamma?
a. Alpha has a 1 to 1 relationship with Gamma. Beta has a 1 to 1 relationship with Gamma.
---b. Alpha has a 1 to * relationship with Gamma. Beta has a 1 to * relationship with Gamma.
c. Alpha has a * to 1 relationship with Gamma. Beta has a * to 1 relationship with Gamma.
d. Alpha has a 1 to 1 relationship with Beta. Beta has a 1 to * relationship with Gamma.
e. Alpha and Beta have no relationship to Gamma.

(26) Which statement correctly characterizes UML models?
---a. You can create UML models at different stages of a project and with different purposes and levels of details
b. You can create UML models at different stages of a project, but they must contain all the low-level details
c. You can only create UML models at the beginning of a project and these models must contain all the low-level details
d. You can only create UML models during the design phase of a project and these models may contain any level of details
e. You can only create UML models during the domain analysis phase of a project

(27) A simple technique for discovering domain classes is to look at a description of requirements and to identify...
a. actors
b. adjectives and descriptive phrases
---c. nouns and noun phrases
d. attributes
e. associations among classes

Chapter 6 -- Using Design Patterns

(28) The Design Pattern used so that objects do not have to change classes is the...
a. Abstraction-Occurrence Pattern
b. Delegation Pattern
c. Immutable Pattern
---d. Player-Role Pattern
e. Singleton Pattern

(29) The Design Pattern used when another class has a method which provides the required service, but inheritance is not appropriate is the...
a. Abstraction-Occurrence Pattern
---b. Delegation Pattern
c. Immutable Pattern
d. Player-Role Pattern
e. Singleton Pattern

(30) The Singleton Pattern contains the following solution...
a. A private class variable, often called theInstance, which stores the single instance
b. A public class method, often called getInstance(). The first time this method is called, it creates the object.
c. Calls after the first to getInstance() simply return theInstance
d. A private constructor
---e. All the above

Chapter 8 -- Modelling Interactions and Behaviour

(31) To model the dynamic aspects of a software system and to show the sequence of messages exchanged by the set of objects performing a certain task, we use....
a. Transition Diagrams
b. State Diagrams
c. Communication Diagrams
---d. Sequence Diagrams
e. Lifeline Diagrams

(32) The start state of a State Diagram is represented by....
a. A white circle
b. A labelled oval
c. A circle with a ring around
---d. A black circle
e. An arrow

(33) In an Activity Diagram a place that has multiple incoming and multiple outgoing transitions is a....
a. swimlane
b. join
c. fork
---d. rendezvous
e. transition point

(34) In an Activity Diagram the partition of activities among existing classes can be explicitly shown using...
---a. swimlanes
b. arrows
c. colors
d. objects
e. comments

Chapter 9 -- Architecting and Designing Software

(35) Why does high system cohesion make it easier to make modifications later?

a. The various aspects of the system are designed so that they can be used again in other contexts
b. It keeps the level of abstraction as high as possible
---c. All code related to the same functionality is kept together
d. Highly cohesive components keep together things that are related to each other along with many other related things
e. High system cohesion implies that several modules call each other

(36) Which of the following violates the Design Principle "Anticipate Obsolescence"?
a. Use standard languages and technologies that are supported by multiple vendors
---b. Use early releases of technology
c. Avoid using software libraries that are specific to particular environments
d. Avoid using undocumented features or little-used features of software libraries
e. Avoid using software or special hardware from companies that are less likely to provide long-term support

(37) Developing a Software Architecture is important for all the reasons below except which one?

a. To enable everyone to better understand the system
b. To allow people to work on individual pieces of the system in isolation
c. To prepare for extension of the system
d. To facilitate reuse and reusability
---e. To better understand what users need and how to create the best user stories

(38) The Service-Oriented Architecture is most like which other architecture?

a. Multi-Layer
---b. Client/Server
c. Transaction-Processing
d. Pipe-and-Filter
e. Model-View-Controller (MVC)

Chapter 7 -- Focusing on Users and Their Tasks

(39) Focusing on users when developing software leads to all the following results except which one?
---a. Greater costs associated with changing the system later
b. Reduced training and support costs
c. Greater efficiency of use
d. Reduced costs by only developing features that are needed
e. Better prioritizing of work for iterative development

(40) If a product provides the capabilities to allow users to achieve their goals, the product is said to have high....
a. adaptability
b. reliability
c. availability
d. usability
---e. utility

(41) User Interfaces are best when based on....
---a. Use cases and user stories
b. Existing systems
c. Project Owner directives
d. Software developer intuition
e. Retired systems

Chapter 10 -- Testing and Inspecting to Ensure High Quality

(42) All the following are examples of Defects in Ordinary Algorithms except which one?
a. Incorrect logical conditions
---b. Deadlock and livelock
c. Not terminating a loop or recursion
d. Not setting up the correct preconditions for an algorithm
e. Not handling null conditions

(43) All the following are examples of Defects in Handling Stress and Unusual Situations except which one?
a. Incompatibility with specific configurations of hardware or software
b. Defects in handling peak loads or missing resources
c. Inappropriate management of resources
---d. Insufficient branch and path testing
e. Insufficient throughput or response time on minimal configurations

(44) How are severity levels expressed for test cases?
a. Level 1: Important cases, Level 2: Unimportant cases
b. Level 1: Daily cases, Level 2: Weekly cases, Level 3: Monthly cases
---c. Level 1: Critical cases, Level 2: General cases, Level 3: Cases of least importance
d. Level 3: Critical cases, Level 2: General cases, Level 1: Cases of least importance
e. Level 0: Critical cases, Level 1: Important cases, Level 2: General cases, Level 3: Cases of least importance

(45) Efforts to fix defects may have actually added new defects. This is the reason for...
a. White box testing
b. Top down testing
c. Bottom-up testing
d. Sandwich testing
---e. Regression testing

(46) There are two variations of Inspections in which the code is described by one of these two....
a. author, recorder
b. moderator, paraphraser
c. moderator, recorder
d. author, moderator
---e. author, paraphraser

Chapter 11 -- Managing the Software Process

(47) What is the typical Agile team organization?
a. PERT team
b. Chief programmer team
c. Hierarchical manager-subordinate team
---d. Egoless/self-organizing team
e. Binary subordinate team

(48) What led to the failure to the classic Waterfall Model?
---a. Did not account for the fact that requirements constantly change
b. Allowed software engineers to move on to the next stage before completing the previous stage
c. Customers could use the system too early, before adequate testing
d. Had no provision for verification and validation
e. Encouraged going back to earlier stages

(49) Which best describes the Iterative nature of Agile development?
a. Development time of each iteration is determined at the end of the previous iteration. The first is always 2 weeks.
b. Development time of earlier iterations is shorter than later iterations
c. Development time of earlier iterations is longer than later iterations
d. Development time of each iteration is short, but may be adjusted as needed
---e. Development time of each iteration is short and the same length

(50) Which of the following does NOT characterize benefits to the Agile project team?

a. Team members collaboratively make decisions and are more empowered
---b. Team members can focus on the entire Product Backlog throughout each Sprint
c. Simple and minimal documents are used
d. All the requirements are not gathered up front and are implemented only when they arise
e. Less time is required for planning