# Module-1 Introduction To Software Engineering & Process Models

## Short Question:
1. Define software.
2. Define software engineering.
3. What do you mean by “software doesn't wear out”?
4. What do you mean by legacy software?
5. Why most software are custom built?
6. Define software process.
7. What do you mean by process model?
8. Which model is named as classic life cycle model and why?
9. Give characteristics of evolutionary process models.
10. Give limitations of evolutionary models.
11. What is PSP and TSP?
12. What do you mean by product and process?
13. What is agility?
14. What is an agile process?
15. List variant of XP?
16. What is prototyping?
17. State the basic objective of spiral model?
18. Define the term unified process.
19. What are scrums?
20. What are sprints?

## Long Question
1. Explain How software is differs from hardware.
2. Explain generic process framework activities.
3. What do you mean by umbrella activities? List any file umbrella activity.
5. Describe the water fall model with its limitations.
6. Describe why water fall model is not realistic?
7. Explain incremental model with their advantages and limitations.
8. What do you mean by prototype? Explain prototyping model.
9. Define the term software process model. Compare the spiral models of software development with prototyping model.
10. Write a short note on CMMI.
11. List principles of agile process.
12. Describe following process model with their advantages and limitations:
   a. Concurrent model.
   b. Component based development
   c. The formal method model
   d. Aspect-Oriented Software Development
   e. The Unified process
13. Explain following agile process:
a. Extreme Programming  
b. Adaptive Software Development  
c. Scrum  
d. Dynamic Systems Development method

14. How is process assessment achieved in case of a software?  
15. What are the advantages of PSP?  
16. How does human factor affect agility?  
17. Write a brief note on DSDM.  
18. What are crystals?  
19. Write a note on feature driven development and its process.  
20. Write a detailed note on process patterns.

Multiple Choice Questions

1. Which question no longer concerns the modern software engineer?  
   a. Why does computer hardware cost so much?  
   b. Why does software take a long time to finish?  
   c. Why does it cost so much to develop a piece of software?  
   d. Why can't software errors be removed from products prior to delivery?

2. Software deteriorates rather than wears out because  
   a. Software suffers from exposure to hostile environments  
   b. Defects are more likely to arise after software has been used often  
   c. Multiple change requests introduce errors in component interactions  
   d. Software spare parts become harder to order

3. Most software continues to be custom built because  
   a. Component reuse is common in the software world.  
   b. Reusable components are too expensive to use.  
   c. Software is easier to build without using someone else's components.  
   d. Off-the-shelf software components are unavailable in many application domains.

4. The nature of software applications can be characterized by their information  
   a. complexity  
   b. content  
   c. determinacy  
   d. both b and c

5. Which of the items listed below is not one of the software engineering layers?  
   a. Process  
   b. Manufacturing  
   c. Methods  
   d. Tools

6. Which of these are the 5 generic software engineering framework activities?  
   a. communication, planning, modeling, construction, deployment  
   b. communication, risk management, measurement, production, reviewing  
   c. analysis, designing, programming, debugging, maintenance  
   d. analysis, planning, designing, programming, testing
7. Process models are described as agile because they
   a. eliminate the need for cumbersome documentation
   b. emphasize maneuverability and adaptability
   c. do not waste development time on planning activities
   d. make extensive use of prototype creation

8. Which of these terms are level names in the Capability Maturity Model?
   a. Performed
   b. Repeated
   c. Reused
   d. Optimized
   e. both a and d

9. Which of these are standards for assessing software processes?
   a. SEI
   b. SPICE
   c. ISO 19002
   d. ISO 9001
   e. both b and d

10. Which of these is not a characteristic of Personal Software Process?
    a. Emphasizes personal measurement of work product
    b. Practitioner requires careful supervision by the project manager
    c. Individual practitioner is responsible for estimating and scheduling
    d. Practitioner is empowered to control quality of software work products

11. Which of these are objectives of Team Software Process?
    a. Accelerate software process improvement
    b. Allow better time management by highly trained professionals
    c. Build self-directed software teams
    d. Show managers how to reduce costs and sustain quality
    e. both b and c

12. The linear sequential model of software development is
    a. A reasonable approach when requirements are well defined.
    b. A good approach when a working program is required quickly.
    c. The best approach to use for projects with large development teams.
    d. An old fashioned model that cannot be used in a modern context.

13. The linear sequential model of software development is also known as the
    a. Classical life cycle model
    b. Fountain model
    c. Spiral model
    d. Waterfall model
    e. both a and d

14. The incremental model of software development is
    a. A reasonable approach when requirements are well defined.
    b. A good approach when a working core product is required quickly.
    c. The best approach to use for projects with large development teams.
    d. A revolutionary model that is not used for commercial products.

15. The rapid application development model is
a. Another name for component-based development.
b. A useful approach when a customer cannot define requirements clearly.
c. A high speed adaptation of the linear sequential model.
d. All of the above.

16. Evolutionary software process models
a. Are iterative in nature
b. Can easily accommodate product requirements changes
c. Do not generally produce throwaway systems
d. All of the above

17. The prototyping model of software development is
a. A reasonable approach when requirements are well defined.
b. A useful approach when a customer cannot define requirements clearly.
c. The best approach to use for projects with large development teams.
d. A risky model that rarely produces a meaningful product.

18. The spiral model of software development
a. Ends with the delivery of the software product
b. Is more chaotic than the incremental model
c. Includes project risks evaluation during each iteration
d. All of the above

19. The concurrent development model is
a. Another name for the rapid application development model.
b. Often used for the development of client/server applications.
c. Only used for development of parallel or distributed systems.
d. Used whenever a large number of change requests are anticipated.

20. The component-based development model is
a. Only appropriate for computer hardware design.
b. Not able to support the development of reusable components.
c. Works best when object technologies are available for support.
d. Not cost effective by known quantifiable software metrics.

21. The formal methods model of software development makes use of mathematical methods
a. Define the specification for computer-based systems
b. Develop defect free computer-based systems
c. Verify the correctness of computer-based systems
d. All of the above

22. Which of these is not one of the phase names defined by the Unified Process model for software development?
a. Inception phase
b. Elaboration phase
c. Construction phase
d. Validation phase

23. Which of the following is not necessary to apply agility to a software process?
a. Eliminate the use of project planning and testing
b. Only essential work products are produced
c. Process allows team to streamline tasks
d. Uses incremental product delivery strategy

24. How do you create agile processes to manage unpredictability?
   a. Requirements gathering must be conducted very carefully
   b. Risk analysis must be conducted before planning takes place
   c. Software increments must be delivered in short time periods
   d. Software processes must adapt to changes incrementally
   e. both c and d

25. Which of the following traits need to exist among the members of an agile software team?
   a. Competence
   b. Decision-making ability
   c. Mutual trust and respect
   d. All of the above

26. What are the four framework activities found in the Extreme Programming (XP) process model?
   a. analysis, design, coding, testing
   b. planning, analysis, design, coding
   c. planning, analysis, coding, testing
   d. planning, design, coding, testing

27. What are the three framework activities for the Adaptive Software Development (ASD) process model?
   a. analysis, design, coding
   b. feasibility study, functional model iteration, implementation
   c. requirements gathering, adaptive cycle planning, iterative development
   d. speculation, collaboration, learning

28. Which is not one of the key questions that is answered by each team member at each daily Scrum meeting?
   a. What did you do since the last meeting?
   b. What obstacles are you encountering?
   c. What is the cause of the problems you are encountering?
   d. What do you plan to accomplish at the next team meeting?

29. Agile Modeling (AM) provides guidance to practitioner during which of these software tasks?
   a. Analysis
   b. Design
   c. Coding
   d. Testing
   e. both a and b

True or False:

1. In general software only succeeds if its behavior is consistent with the objectives of its Designers.
2. Most software development projects are initiated to try to meet some business
need.
3. Change cannot be easily accommodated in most software systems, unless the system was designed with change in mind.
4. The functionality of most computer systems does not need to be enhanced the lifetime of the system.
5. The so called "new economy" that gripped commerce and finance during the 1990s died and no longer influences decisions made by businesses and software engineers.
6. Modern software applications are so complex that it is hard to develop mutually exclusive category names.
7. Software is a product and can be manufactured using the same technologies used for other engineering artifacts.
8. Today the increased power of the personal computer has brought about an abandonment of the practice of team development of software.
9. Process technology tools allow software organizations to compress schedules by skipping unimportant activities.
10. It is generally accepted that one cannot have weak software processes and create high quality end products.
11. The best software process model is one that has been created by the people who will actually be doing the work.
12. Software processes can be constructed out of pre-existing software patterns to best meet the needs of a software project.
13. Software engineering umbrella activities are only applied during the initial phases of software development projects.
14. In the Unified Process model requirements are determined iteratively and may span more than one phase of the process.
15. It is not possible to build software that meets the customers’ needs today and exhibits the quality characteristics that will enable it to be extended tomorrow.
16. In agile software processes the highest priority is to satisfy the customer through early and continuous delivery of valuable software.
17. Agility is nothing more than the ability of a project team to respond rapidly to change.
18. All agile process models conform to a greater or lesser degree to the principles stated in the "Manifesto for Agile Software Development".
19. The Dynamic Systems Development Method (DSDM) suggests a philosophy that is based on the Pareto principle (80% of the application can be delivered in 20% of the time required to build the complete application).
20. In Feature Driven Development (FDD) a "feature" is a client-valued function that can be delivered in two months or less.

Fill in the blanks:
1. _____ is dose not wear out.
2. Software is ____ and ____ at the same time.
3. Software is engineered but not ____.
4. _____ Stage Software failure rate is high.
5. _____ Types of software approach.
6. Design is a _____ software engineering activity.
7. Software engineering defines communication, planning, modeling _____ and _____
8. Project management is a ____ and ____ initial concept through to system activity.
9. Products that are reliable, robust _____ flexible and maintainable.
10. Water fall model sometimes called____
11. The Increment model combines elements of ------- and _____ process flow sequence.
12. ------- is risk driven process model.
13. ------- Model encompasses a set of activates that leads to formal intentional specification of computer software.
14. An agile Process reduces--------.
15. Adaptive software Development has been proposed by______.
16. The information obtained from the customer during inception and elicitation is-------
   -- and _____- during elaboration.
17. A UML activity diagram represents the ------- and ------- that occur as some function is performed.
18. ------- define the properties of a data object.
19. Software cannot ____ but can be _____
20. Umbrella activities are ____ for process framework

Module 2: Requirement Analysis

Short Questions
1. What is software engineering practice?
2. List tasks of requirement engineering.
3. What is QFD?
4. What are the types of requirement according to QFD?
5. List work product of requirement elicitation.
6. What are the different ways of building the requirement model?
7. What is analysis pattern?
8. What do you mean by validation and verification?
9. What does win-win means in the context of negotiation during the requirement engineering activity?
10. What is product and process?
11. What is the fundamental difference between the structured analysis and object-oriented strategies for requirement analysis?
12. What is represented by circle in DFD?
13. What is process activation table? Give its purpose.
14. What is process specification?
15. Are PSPEC and use case are same thing? If not, explain difference.
16. How does sequence diagram and state diagram are different? How are they similar?
17. What is the purpose of the content model for a WebApp?
18. What is the purpose of a configuration model for WebApp?
19. Why there arose essentials of software engineering practice?
20. List the steps involved in planning practices.

Long Questions
1. Describe core principles that guide process. Which do you believe is most important?
2. List and describe the principles that guide the practice.
3. Describe communication principles.
4. Describe planning principles.
5. Describe requirement modeling principles.
6. Describe design modeling principles.
7. Describe construction principles.
8. Describe deployment principles.
9. Describe testing principles.
10. Of the eight core principles that guide practice, which do you believe is most important?
11. How does agile communication differ from traditional software engineering communication? How is it similar?
12. Explain why project planning needs to be iterative.
13. Why are models important in software engineering work? Are they always necessary? Are there qualifiers to your answer about necessity?
14. What three characteristics are considered during design modeling?
15. Explain why exhaustive testing cannot be used to prove software is error free.
16. Why is feedback important to the software team?
17. What is requirement engineering? Describe the seven distinct tasks encompassed in requirement engineering.
18. Develop a complete use case for following activities:
   a. Withdrawal at an ATM
   b. Deposit amount at ATM
   c. Buying stock using an online account
   d. Issue book using online library
   e. To submit assignment online
   f. To pay fees at institute
   g. To access email account
   h. To send an email
19. Develop DFD for following system:
   a. Online course registration system for University
   b. Web-Based order-processing system
   c. Billing system for restaurant
   d. Online system for senior citizen (make your assumption)
20. Describe the elements of requirements model.
21. Software is to be developed for hotel management system in which information is provided for all type of activities conducted in hotel. The major users of the system are hotel staff, people who stay in the hotel and people who visit the restaurant. Information for the billing system, hotel account management, staff salary, hotel menu information, hotel room information is provided by software. Prepare Software Requirement Specification and Use Case Diagram.
22. A Library lends books and magazines to member, who is registered in the system. Also
it handles the purchase of new titles for the Library. Popular titles are bought into multiples copies. Old books and magazines are removed when they are out of date or in poor condition. A member can reserve a book or magazine that is not currently available in the library, so that when it is returned or purchased by the library, that person is notified. The library can easily create, replace and delete information about the titles, members, loans and reservation in the system. Prepare Software Requirement Specification and Use Case Diagram.

### Multiple Choice Questions

1. Which of the following is not one of Hooker’s core principles of software engineering practice?
   - A) All design should be as simple as possible, but no simpler
   - B) A software system exists only to provide value to its users.
   - C) Pareto principle (20% of any product requires 80% of the effort)
   - D) Remember that you produce others will consume

2. Software engineers collaborate with customers to define which of the following?
   - A) Customer visible usage scenarios
   - B) Important software features
   - C) System inputs and outputs
   - D) All of the above

3. Everyone on the software team should be involved in the planning activity so that we can
   - A) reduce the granularity of the plan
   - B) analyze requirements in depth
   - C) get all team members to "sign up" to the plan
   - D) begin design

4. What role(s) do user stories play in agile planning?
   - A) Define useful software features and functions delivered to end-users
   - B) Determine a schedule used to deliver each software increment
   - C) Provide a substitute to performing detailed scheduling of activities
   - D) Used to estimate the effort required to build the current increment
   - E) both a and d

5. Which of the following activities is not one of the four things that need to be accomplished by the generic planning task set?
   - A) Develop overall project strategy
   - B) Identify the functionality to deliver in each software increment
   - C) Create a detailed schedule for the complete software project
   - D) Devise a means of tracking progress on a regular basis

6. Analysis models depict software in which three representations?
   - A) architecture, interface, component
   - B) cost, risk, schedule
   - C) information, function, behavior
   - D) None of the above

7. Which of the following is not one of the principles of good coding?
   - A) Create unit tests before you begin coding
   - B) Create a visual layout that aids understanding
C) Keep variable names short so that code is compact
D) Write self-documenting code, not program documentation

8. Which of the following are tasks in the generic task set for construction?
   A) Build a software component
   B) Create a user interface
   C) Unit test the component
   D) Assess the quality of the component
   E) both a and c

9. Which of the following are valid reasons for collecting customer feedback concerning delivered software?
   A) Allows developers to make changes to the delivered increment
   B) Delivery schedule can be revised to reflect changes
   C) Developers can identify changes to incorporate into next increment
   D) All of the above

10. During project inception, the intent of the tasks is to determine
    A) basic problem understanding
    B) nature of the solution needed
    C) people who want a solution
    D) none of the above
    E) a, b, and c

11. Three things that make requirements elicitation difficult are problems of
    A) budgeting
    B) scope
    C) understanding
    D) volatility
    E) b, c, and d

12. The result of the requirements engineering elaboration task is an analysis model that defines which of the following problem domain(s)?
    A) information
    B) functional
    C) behavioral
    D) all of the above

13. The system specification describes the
    A) Function, performance and constraints of a computer-based system
    B) implementation of each allocated system
    C) element software architecture
    D) time required for system simulation

14. The best way to conduct a requirements validation review is to
    A) examine the system model for errors
    B) have the customer look over the requirements
    C) send them to the design team and see if they have any concerns
    D) use a checklist of questions to examine each requirement

15. The use of traceability tables helps to
    A) debug programs following the detection of run-time errors
    B) determine the performance of algorithm implementations
C) identify, control, and track requirements changes
D) none of the above

16. Which of the following is not one of the context-free questions that would be used during project inception?
   A) What will be the economic benefit from a good solution?
   B) Who is against this project?
   C) Who will pay for the work?
   D) Who will use the solution?

17. In collaborative requirements gathering, the facilitator
   A) cannot be a member of the software team
   B) cannot be a customer
   C) controls and facilitates the process
   D) must be an outsider

18. Which of the following is not one of the requirement classifications used in Quality Function Deployment (QFD)?
   A) exciting
   B) expected
   C) mandatory
   D) normal

19. The work products produced during requirement elicitation will vary depending on the
   A) size of the budget
   B) size of the product being built
   C) software process being used

20. Which of following is not a UML diagram used creating a system analysis model?
   A) activity diagram
   B) class diagram
   C) dataflow diagram
   D) state diagram

21. Which of the following is not an objective for building an analysis model?
   A) define set of software requirements that can be validated
   B) describe customer requirements
   C) develop an abbreviated solution for the problem
   D) establish basis for software design

22. The data dictionary contains descriptions of each software
   A) control item
   B) data object
   C) diagram
   D) notation
   E) both a and b

23. Which of these is not an element of an object-oriented analysis model?
   A) Behavioral elements
   B) Class-based elements
   C) Data elements
   D) Scenario-based elements

24. The relationships shown in a data model must be classified to show their
25. The entity relationship diagram
   A) depicts relationships between data objects
   B) depicts functions that transform the data flow
   C) indicates how data are transformed by the system
   D) indicates system reactions to external events
   E) both a and c

26. A generalized description of a collection of similar objects is a
   A) class
   B) instance
   C) subclass
   D) super class

27. UML activity diagrams are useful in representing which analysis model elements?
   A) Behavioral elements
   B) Class-based elements
   C) Flow-based elements
   D) Scenario-based elements

28. The data flow diagram
   A) depicts relationships between data objects
   B) depicts functions that transform the data flow
   C) indicates how data are transformed by the system
   D) indicates system reactions to external events
   E) both b and c

29. Control flow diagrams are
   A) needed to model event driven systems.
   B) required for all systems.
   C) used in place of data flow diagrams.
   D) useful for modeling real-time systems.
   E) both a and d

30. Which of the following should be considered as candidate objects in a problem space?
   A) events
   B) people
   C) structures
   D) all of the above

31. Which of the following is not one of the broad categories used to classify operations?
   A) computation
   B) data manipulation
   C) event monitors
   D) transformers

32. Which of the following items does not appear on a CRC card?
   A) class collaborators
   B) class name
C) class reliability
D) class responsibilities

True or False
1. The essence of software engineering practice might be described as understand the problem, plan a solution, carry out the plan, and examine the result for accuracy.
2. Every communication activity should have a facilitator to make sure that the customer is not allowed to dominate the proceedings.
3. The agile view of iterative customer communication and collaboration is applicable to all software engineering practice.
4. The customer can directly observe both the difference between the internal quality of a design and its external quality?
5. Teams using agile software practices never create models.
6. Many of the tasks from the generic task sets for analysis modeling and design can be conducted in parallel with one another.
7. A successful test is one that discovers at least one as-yet undiscovered error.
8. Requirements engineering is a generic process that does not vary from one software project to another.
9. It is relatively common for different customers to propose conflicting requirements, each arguing that his or her version is the right one.
10. A stakeholder is anyone who will purchase the completed software system under development.
11. The job of the requirements engineer is to categorize all stakeholder information in a way that allows decision makers to choose an internally consistent set of requirements.
12. The nature of collaboration is such that all system requirements are defined by consensus of a committee of customers and developers.
13. Developers and customers create use-cases to help the software team understand how different classes of end-users will use functions.
14. Analysis patterns facilitate the transformation of the analysis model into a design model by suggesting reliable solutions to common problems.
15. Use-case actors are always people, never system devices.
16. In win-win negotiation, the customer's needs are met even though the developer's need may not be.
17. In requirements validation the requirements model is reviewed to ensure its technical feasibility.
18. Object-oriented domain analysis is concerned with the identification and specification of reusable classes within an application domain.
19. In analysis models the only data objects that need representation are those that will be...
implemented using software classes.
20. The values that are assigned to an object's attributes make that object unique.
21. Operations are object procedures that are invoked when an object receives a message.
22. In many cases there is no need to create a graphical representation of a usage scenario.
23. The data flow diagram must be augmented by descriptive text in order to describe the functional requirements for a software product.
24. Attributes cannot be defined for a class until design has been completed.
25. An analysis package involves the categorization of analysis model elements into useful groupings.

Module 3 Object Oriented Modeling Concepts

Short Questions
1. Draw a UML diagram to represent a class and its object
2. Show how attribute values are represented in a class model
3. Represent a function (operation) in a class along with its signature.
4. Show many-to-many association between classes Person and Company.
5. Represent one-to-one association between Class Country and CapitalCity.
6. Assign association end name for classes Person and Company with an association WorksFor.
7. Create an Association class between File and User class
8. Which type of association is shown in the diagram below

![Diagram](image)

9. Create enumerations for class Movie
10. List which multiplicity attributes are show in the class below

![Class Diagram](image)

11. Draw a UML to show ternary association between Professor, Semester, ListedCourse.
12. Prepare a list of classes that you would expect each of the following systems to handle.
   a. A program for laying out a newspaper
   b. A program to compute and store bowling scores
13. What is a model?
14. List the purpose for which a model is used.
15. What is an object? Give example.
16. What is the basic purpose of class diagram?
17. What are values and attributes?
18. Give an example to signify the concept of operation.
19. What is signature?
20. Give few examples of Feature.
21. What are links and associations?
22. What are association end names?
23. What are association classes?
24. Differentiate bags and sequences with an example.
25. With the help of an UML notation define the term signal event?
26. Define the term transition with an example and with a UML notation.
27. What are entry activities give an example as well as represent it in UML form?

### Long Questions

1. Explain class model with appropriate examples.
2. Describe the state model with an illustration.
3. How is interaction model different from state model?
4. Write a note on multiplicity of attributes.
5. Write a short note on generalization and inheritance in OOAD.
6. State the importance of state modeling for a software system.
7. What are N-ary associations?
8. Explain the concept of abstract classes with the help of UML notation.
9. What are constraints? Where are these constraints used? Explain with the help of a diagram.
10. With the help of a notation explain the concept of package.
11. Differentiate various types of events with an example for each.
12. Write a short note on States.
13. Describe the behavior of state diagram with valid examples.
14. Draw a sequence diagrams for a stock purchase system with all its assumptions.
15. List and explain the various guidelines for sequence models.
16. List and explain the varying guidelines for use case relationships with an example.
17. Write a short note on Activity diagram.
18. What are workarounds?
19. Write a short note on qualified associations.
20. Draw a state diagram for the game of Bridge.
21. Given below is the state diagram for a Tennis game:
   a. List all states of the state diagram below.
   b. List all the types of events of the state diagram.
   c. List the guard conditions if any.
   d. List the transitions of the state diagram
   e. List the activity effects if any
Multiple Choice Questions

1. A description of the real-world objects reflected within the systems
   a. Domain model
   b. Application model
   c. Class model
   d. All of the model

2. The ___ describes the static structure of the objects in a system and their relationships.
   a. State model
   b. Class model
   c. Interaction model
   d. All of the model

3. The ___ describes the aspects of an object that change over time.
   a. State model
   b. Class model
   c. Interaction model
   d. All of the model

4. Divide a complex system into small, self-contained pieces that can be managed independently. How is it called?
   a. Abstraction
   b. Modularity
   c. Encapsulation
   d. Hierarchy

5. In order to model the relationship “a course is composed of 5 to 20 students and one or more instructors”, you could use.
   a. Aggregation
   b. Composition
   c. Association
   d. Realization

6. Which of the following is not an objective for building an analysis model?
a. Define set of software requirements that can be validated
b. Describe customer requirements
c. Develop an abbreviated solution for the problem
d. Establish basis for software design

7. Which of these is not an element of an object oriented analysis model?
   a. Behavioral elements
   b. Class-based elements
c. Data elements
d. Scenario-based elements

True False
1. Abstraction lets you focus on essential aspects of an application while ignoring details.
2. Encapsulation separates the external aspects of an object that are accessible to other objects.
3. Object-oriented domain analysis is concerned with the identification and specification of reusable classes within an application domain.

Fill in the blanks
1. A ___ focuses on the functionality of a system that is what a system does for users.
2. A ___ shows the objects that interact and the time sequence of their interactions.
3. An ___ diagram elaborates important processing steps.
4. A ___ is an abstraction of something for the purpose of understanding it before building it.

Module 4 Software Design

Short Questions
1. Define Archetype.
2. What are the attributes of good software?
3. List elements of design model.
4. Give purpose of software design.
5. List quality attributes of good software design.
6. What is architecture?
7. What do you mean by subordinate and superordinate system?
8. What is peers?
9. How target systems are represented in architecture?
10. What is node and detector?
11. What is component?

Long Questions
1. Explain following design concept in detail: Abstraction, architecture, patterns, separation of concerns, modularity, functional independence, refinement, aspect and refactoring
2. List the fundamental design concepts. Discuss modularity and functional independence.
3. What is software architecture? List different architecture styles. Discuss call return
architectural style.
4. Describe different types of design classes according to object oriented design.
5. Explain characteristics of good class design.
6. Explain design model in detail.
7. Explain following architecture style in detail:
   Data centered architecture, data flow architecture, call and return architecture, object
   oriented architecture and layered architecture
8. What is data design?

Multiple Choice Questions
1. Which of the following are areas of concern in the design model?
   A) architecture
   B) data
   C) interfaces
   D) project scope
   E) a, b and c
2. The importance of software design can be summarized in a single word
   A) accuracy
   B) complexity
   C) efficiency
   D) quality
3. Which of these are characteristics of a good design?
   A) exhibits strong coupling between its modules
   B) implements all requirements in the analysis model
   C) includes test cases for all components
   D) provides a complete picture of the software
   E) both b and d
4. Which of the following is not a characteristic common to all design methods?
   A) configuration management
   B) functional component
   C) notation quality assessment
   D) guidelines refinement heuristics
5. What types of abstraction are used in software design?
   A) control
   B) data
   C) environmental
   D) procedural
   E) a, b and d
6. Which of the following models can be used to represent the architectural design of a piece
   of software.
   A) Dynamic models
   B) Functional models
   C) Structural models
   D) All of the above
7. Cohesion is a qualitative indication of the degree to which a module
A) Can be written more compactly.
B) Focuses on just one thing.
C) is able to complete its function in a timely manner.
D) is connected to other modules and the outside world.

8. Coupling is a qualitative indication of the degree to which a module
   A) Can be written more compactly.
   B) Focuses on just one thing.
   C) is able to complete its function in a timely manner.
   D) Is connected to other modules and the outside world.

9. Polymorphism reduces the effort required to extend an object system by
   A) coupling objects together more tightly.
   B) enabling a number of different operations to share the same name.
   C) making objects more dependent on one another.
   D) removing the barriers imposed by encapsulation.

10. Which of the following is not one of the five design class types
    A) Business domain classes
    B) Entity classes
    C) Process classes
    D) User interface classes

11. Which design model elements are used to depict a model of information represented from
    the user's view?
    A) Architectural design elements
    B) Component-level design elements
    C) Data design elements
    D) Interface design elements

12. Which design is analogous to the floor plan of a house?
    A) Architectural design
    B) Component-level design
    C) Data design
    D) Interface design

13. Which design model is analogous to the detailed drawings of the access points and external utilities for a house?
    A) Architectural design
    B) Component-level design
    C) Data design
    D) Interface design

14. Which design model is analogous to a set of detailed drawings for each room in a house?
    A) Architectural design
    B) Component-level design
    C) Data design
    D) Interface design

15. Which of these characteristics are true of a data warehouse, but not a typical database?
    A) business level orientation
B) currency of information
C) integration
D) nonvolatility
E) both c and d
16. An architectural style encompasses which of the following elements?
   A) constraints
   B) set of components
   C) semantic models
   D) syntactic models
   E) a, b and c
17. To determine the architectural style or combination of styles that best fits the proposed system, requirements engineering is used to uncover
   A) algorithmic complexity
   B) characteristics and constraints
   C) control and data
   D) design patterns
18. The criteria used to assess the quality of an architectural design should be based on system
   A) accessibility
   B) control
   C) data
   D) implementation
   E) both b and c
19. During the process of modeling the system in context, systems that interact with the target system are represented as
   A) Peer-level systems
   B) Subordinate systems
   C) Superordinate systems
   D) Working systems
   E) a, b and c
20. Which of the following is not an example of infrastructure components that may need to be integrated into the software architecture?
   A) Communications components
   B) Database components
   C) Interface components
   D) Memory management components
21. In the architecture trade-off analysis method the architectural style should be described using the
   A) data flow view
   B) module view
   C) process view
   D) user view
22. A useful technique for evaluating the overall complexity of a proposed architecture is to look at the component:  
   A) cohesion flow  
   B) dependencies  
   C) sharing dependencies  
   D) size  
   E) both b and c  

23. When the overall flow in a segment of a data flow diagram is largely sequential and follows straight-line paths, _______ is present.  
   A) low coupling  
   B) good modularity  
   C) transaction flow  
   D) transform flow  

24. When a single item that triggers other data flow along one of many paths of a data flow diagram, _______ characterizes the information flow.  
   A) high coupling  
   B) poor modularity  
   C) transaction flow  
   D) transform flow  

25. In transaction mapping the first level factoring results in the  
   A) creation of a CFD  
   B) derivation of the control hierarchy  
   C) distribution of worker modules  
   D) refinement of the module view  

26. A successful application of transform or transaction mapping to create an architectural design is supplemented by  
   A) entity relationship diagrams  
   B) module interface descriptions  
   C) processing narratives for each module  
   D) test cases for each module  
   E) both b and c  

True False  
1. Software design is an iterative generic process that may be applied without modification to any software project.  
2. Design patterns are not applicable to the design of object-oriented software?  
3. Since modularity is an important design goal it is not possible to have too many modules in a proposed design.  
4. Information hiding makes program maintenance easier by hiding data and procedure from unaffected parts of the program.
5. When using structured design methodologies the process of stepwise refinement is unnecessary.
6. Software designs are refactored to allow the creation of software that is easier to integrate, easier to test, and easier to maintain.
7. Inheritance provides a mechanism by which changes to lower level classes can be propagated to all super classes quickly.
8. The deployment design elements specify the build order for the software components.
9. One of the key problems in software reuse is the inability to find existing reusable design patterns when hundreds of candidates exist.
10. Design patterns are best thought of as coding patterns.
11. Frameworks and design patterns are the same thing as far as designers are concerned.
12. The best representation of system architecture is an operational software prototype.
13. The architectural representations can be an enabler for communication among project stakeholders.
14. Data design actually begins during the creation of the analysis model, not the architectural model.
15. Before an architectural pattern can be chosen for use in a specific system it must have a code implementation to facilitate its reuse.
16. Once selected, archetypes always need to be refined further as architectural design proceeds.
17. Quantitative methods for assessing the quality of proposed architectural designs are readily available.
18. When you encounter both transform flow and transaction flow in the same DFD the flow is partitioned and the appropriate mapping technique is used on each part of the DFD.
19. In the most general sense a component is a modular building block for computer software.
20. Software engineers always need to create components from scratch in order to meet customer expectations fully.