Unit 1

1) The reasons behind modeling can be

Readability
Reusability
Both
None

Answer Both

2) The Unified Modeling Language (UML) is a standard language for

specifying
visualizing
constructing
all

Answer all

3) The primary goals in the design of the UML were

security
interactivity
both
none

Answer none

4)Total valid UML diagrams

7
8
9
10

Answer 9

5) UML Diagram Classification

Static, Dynamic
0-1
small, large

no option

Answer: Static, Dynamic

6) Modeling is a mean for dealing with complexity

TRUE
FALSE
partial
dont know

Answer: true

7) Class diagrams represent the structure of the system

TRUE
FALSE
not sure
dont know

Answer: true

8) A class represent a concept

TRUE
FALSE
not sure
dont know

Answer: true

9) An activity diagram dont shows flow control within a system

TRUE
FALSE
partial
not sure

Answer: False

10) Programmers Approach to Software Engineering

Skip requirements engineering and design phases
start writing code
above two
none

Answer: above two

11) Design is a waste of time

programmers approach
Designer Approach
Engineer's Approach
Manager Approach

Answer: programmers approach

12) We need to show something to the customer real quick
Programer's Approach
Designer Approach
Engineers Approach
Manager's Approach

Answer: Manager's approach

13) Design is a trial-and-error process

TRUE
FALSE
not sure
dual

Answer: not sure or true

14) Software design as a wicked problem

TRUE
FALSE
not sure
dont know

Answer: true

16) Every wicked problem is a symptom of another problem

TRUE
FALSE
none
both

Answer: true

17) Following is a design principle

class
Abstraction
Polymorphism
Inheritance

Answer: class (not sure)

18) Complexity

High value is high complexity
Low value is high complexity
Zero value is high complexity
High value is no complexity

Answer: (Didn't get what the question wants to ask)

19) Design methods
jsp
jsd
er
all

Answer: All

20) OOD methods
fusion
booch
both
none

Answer: Both

21) JSP is
Jackson Structured Programming
Jackson Structured Project
both
none

Answer: None

22) Which programming language is the foundation of the Jackson Library?
Jackson
J2EE
both
none

Answer: None

23) JSD is
Jackson Structured Data
Jackson Structured Design
both
none

Answer: Jackson Structured Design

24) JSP is for
programming-in-the-small
programming-in-the-large
both
none

Answer: Both

25) JSD is for
programming-in-the-small
programming-in-the-large
both
none

26) Does Jackson support data binding?
yes
no

Answer: yes

27) JSP basic idea is
bad program reflects structure of its input and output
program reflects structure
good program reflects structure of its input
good program reflects structure of its input and output

Answer: good program reflects structure of its input and output

28) In jsp, program can be derived almost mechanically from a description of the input and output
TRUE
FALSE

Answer: True

29) input and output are depicted in a structure diagram and/or in structured text/schematic logic this is concept of--
jsp
jsd
both
none

Answer: both

30) Basic compound forms of jsp is/are
sequence
iteration
selection
all

Answer: all

31) Model input and output in jsp
use
any diagrams
State diagrams
structure diagrams
class diagrams

Answer: structure diagrams

32) In JSP, Merge diagrams to
create program structure
create object structure
create class structure
all

Answer: WTF Questions wants to ask

33) IN JSP, Optimize results through
program inversion
Simple optimization
both
none

Answer: program inversion

34) The modeling stage, network stage, implementation stage are stages of
JSP
JSD
both
none

Answer: JSD

35) How many ways does Jackson provide to process JSON?
5
4
3
2

Answer: 3

36) JSD life cycle is depicted as
process structure diagram
program structure diagram
Project structure diagram
Part of structure diagram

Answer:

37) Is there any additional library required by the Jackson library outside the JDK?
Yes

No

Answer: No

38) process structure diagrams are
finite state diagrams
infinite state diagrams
state diagrams
Interstate diagrams

Answer: State Diagram

39) identify the objects, determine their attributes and services, determine the relationships between objects are stages of
JSP
JSD
OOAD
OOD

Answer: OOAD

40) Software Life Cycle Activities, in Requirements Specification
System analyst works with users to clarify the detailed system requirements
System manager works with users to clarify the detailed system requirements
System leader works with users to clarify the detailed system requirements
System tester works with users to clarify the detailed system requirements

Answer: System analyst works with users to clarify the detailed system requirements

41) Is Jackson library open-source?
yes
no

Answer: YES

42)Software Life Cycle Activities, in Analysis
Make sure you partially understand the problem before starting the design or program a solution
Make sure you completely understand the problem before starting the design or program a solution
Make sure you completely understand the problem before end the design or program a solution
Make sure you completely understand the problem before starting the analysis or program a solution

Answer: Make sure you completely understand the problem before starting the design or program a solution

43) Software Life Cycle Activities, in Design
Top-down: break system into larger main system
Top-down: combine system into smaller subsystems
Top-down: break system into smaller subsystems
Top-down: combine system into larger system

Answer: Top-down: break system into smaller subsystems

44) The Unified Modeling Language (UML) is a standard language for
specifying
visualizing
constructing
all

Answer: All

45) The reasons behind modeling can be
Readability Reusability both none

Answer: Both

46) The primary goals in the design of the UML were
Security
Interactivity
both
none

Answer:

47) Total valid UML diagrams
7
8
9
10

Answer: 9

48) UML Diagram Classification

Static, Dynamic
True, False
Small, large
0-1

Answer: Static, Dynamic

49) Modeling is a mean for dealing with complexity
TRUE
FALSE
Partial
Don’t know

Answer: True

50) Class diagrams represent the structure of the system.
TRUE
FALSE
Partial
Don’t know

Answer: true

51) How many types of data binding does Jackson support?
5 4 3 2

Answer: 2

52)A class represent a concept.
TRUE FALSE Partial Don’t know

Answe: true

53)An activity diagram dont shows flow control within a system
TRUE FALSE Partial Don’t know

Answer: true

Unit 2

54) All architecture is software design, but not all design is software architecture

TRUE FALSE not sure no option

Answer: true

55) Architecture focuses on â€˜issues that will be difficult/impossible to change once the system is builtâ€™
TRUE FALSE not sure no option

Answer: true

56) Architecture is the fundamental organization
A. of a system, embodied in its components.
B. A and their relationships to each other and the environment
C. B and the principles governing its design and evolution.
D. Nothing like it

Answer: C

57) Data passing mechanisms
Function call System call both none

Answer: Function call

58) Control flow is
Sequential Concurrent both none

Answer Both

59) Non-functional requirements (NFRs) include
Technical Constraints
Bussiness Constraints
qos
all

Answer: All

60) What does an Architect do
Liaison with stakeholders
Technology knowledge
Risk managements
all

Answer: all

61) What are Quality Attributes
reliability
smartness
both
none

Answer: reliability

62) Throughput is
Performance
Complexity
security
none

Answer: Performance (not sure)

63) Security is
performance
QoS
Complexity
Part of reliability

Answer: Part of reliability (not sure)

64) Non-functional requirements are also called as
QOS
Feedback
nob qos
ALL

Answer: QOS

65) Control flow can be
Synchronous
Non-Synchronus
Both
none

Answer: both

66) Patterns Help efficiently communicate a design
TRUE FALSE not sure no option

Answer: true

67) Patterns and Styles are not the same thing
TRUE FALSE not sure no option

Answer: true

68) Non-functional requirements (NFRs) do not define â€˜howâ€™ a system works
TRUE FALSE not sure no option

Answer: true

69) Architecture provides an abstract view of a design by
Hides complexity of design
direct mapping between architecture elements and software elements
both
none

Answer: Hides complexity of design

70) Hierarchical decomposition is a powerful abstraction mechanism
TRUE FALSE not sure no option

Answer: true

71) A software architecture represents a simple design artifact
TRUE FALSE Not sure no option

Answer: false

72) Process view: describes the concurrency and communications elements of architecture.
2+1 View Model 3+1 View Model 4+1 View Model 5+1 View Model

Answer: 4+1 View Model

72) Logical view: describes architecturally significant elements of the architecture and the relationships between them.
2+1 View Model 3+1 View Model 4+1 View Model 5+1 View Model

Answer: 4+1 View Model

73) The design process for identifying the sub-systems making up a system and the framework

for sub-system control and communication is
architectural design
Software design
Data design
Process design

Answer: architectural design

74) The output of this design process is a description of the---
software architecture
data architecture
both
none

Answer: Both

75) An early stage of the system design process
Data design
Software design
architectural design
None of above

Answer: architectural design

76) What Represents the link between specification and design processes
Data design
Software design
architectural design
None of above

Answer: architectural design

77) What involves identifying major system components and their communications
Data design
Software design
architectural design
None of above

Answer: architectural design

78) The system is decomposed into several principal sub-systems and communications between these sub-systems are identified as ---
System structuring

Control modelling
Modular decomposition
None of above

Answer: System structuring

79) A model of the control relationships between the different parts of the system is established as ---
System structuring
Control modelling
Modular decomposition
None of above

Answer:Control modelling

80) The identified sub-systems are decomposed into modules as ---
System structuring
Control modelling
Modular decomposition
None of above

Answer: Modular decomposition

81) A -------- is a system in its own right whose operation is independent of the services provided by other sub-systems
Sub system
Super system
Co system
System of system

Answer: Sub system

82) A ------is a system component that provides services to other components but would not normally be considered as a separate system
Co-module
module
Sub-module
None of above
Modular

Answer: module

83) Different architectural models may be produced during the ----

design process
Engineering process
-
A

Answer: design process

84) Each model presents which different perspectives on the architecture
Static structural model
Dynamic process model
Interface model
All of above

Answer: all

85) -------that shows the major system components
Static structural model
Dynamic process model
Interface model
-

Answer: Static structural model

86) ------that shows the process structure of the system
Static structural model
Dynamic process model
Interface model
-

Answer: Dynamic process model

87) -----that defines sub-system interfaces
Static structural model
Dynamic process model
Interface model
-

Answer: Interface model

88) Uses of distributed object architecture is
As a logical model that allows you to structure and organise the system.
As a non flexible approach to the implementation of client-server systems.
As a physical model that allows you to structure and organise the system.

As a view model that allows you to structure and organise the system.

Answer: As a logical model that allows you to structure and organise the system

89) Advantages of distributed object architecture
It allows the system designer to delay decisions on where and how services should be provided
It is a very open system architecture that allows new resources to be added to it as required
The system is flexible and scaleable
All of above

Answer: all

90) Which of following is true for Distributed object architectures
Each distributable entity is an not object
There is no distinction in a distributed object architectures between clients and servers
Object communication is through a non middleware system
Simplest to design than C/S systems

Answer: 😵

91) Three-tier architectures are
In a three-tier architecture, each of the application architecture
layers may execute on a separate processor
Allows for better performance than a thin-client approach and is simpler to manage than a fat-client approach
A more scalable architecture - as demands increase, extra servers can be added
All of above

Answer: all

92) More processing is delegated to the client as the application processing is locally executed
Flat client model
Thin client model
Thin server model
Flat server model

Answer: Thin client model

93) Most suitable for new C/S systems where the capabilities of the client system are known in advance
Flat client model
Thin client model
Thin server model

Flat server model

Answer: Thin client model

94) How do architects influence on developing organization?
Long term business
Immediate business
Organization structure
All of the above

Answer: All of the above

95) Which of the following factors are influenced on the architect?
Background and experience of the architects
Developing an organization
Customers and end users
All of the above

Answer: All of the above

96) More complex than a thin client model especially for management.
Fat client model
Thin client model
Thin server model
Fat server model

Answer: Flat client model

97) Used when legacy systems are migrated to client server architectures
Fat client model
Thin client model
Thin server model
Fat server model

Answer: Fat client model

98) A major disadvantage of ----- is that it places a heavy processing load on both the server and the network
Fat client model
Thin client model
Thin server model
Fat server model

Answer: Thin client model

99) ----- Concerned with presenting the results of a computation to system users and with collecting user inputs
Application processing layer
Data management layer
Presentation layer
None of above

Answer: Presentation layer

100) --- Concerned with providing application specific functionality
Application processing layer
Data management layer
Presentation layer
None of above

Answer: Application processing layer

101) Distribution of process to processor may be pre-ordered or may be under the control of a dispatcher
Multiprocessor architectures
Single processor architectures
Non-processor architectures
Nano processor architectures

Answer:

102) Which one is true with regards to the architecture business cycle?
The architecture affects the structure of developing organizations
The architecture can affect the enterprise goals of the developing
All of the Above
None of the these

Answer:

103) System composed of multiple processes which may (but need not) execute on different processors
Single processor architectures
Multiprocessor architectures
Non-processor architectures
Nano processor architectures

Answer:

104) Architectural model of many large real-time systems is part of
Single processor architectures
Non-processor architectures
Nano processor architectures
None of above

Answer:

Unit 3

105) Architectural Patterns are
Related to large-scale and coarse-grained design
Related to small-scale and coarse-grained design
both
none

Answer: Related to large-scale and coarse-grained design

106) Architectural Patterns are
applied during the early iterations
applied during the post iterations
both
none

Answer: applied during the early iterations

107) Design Patterns are
small and medium-scale design of objects and frameworks
large and medium-scale design of objects and frameworks
both
none

Answer: small and medium-scale design of objects and frameworks

108) Design Patterns are
Applicable to designing a solution for connecting the small scale elements
Applicable to designing a solution for connecting the large scale elements
both
none

Answer: both

109) Design Patterns are Done during detailed design work after architectural design is â€˜solid.â€™
TRUE FALSE partially true partially false

Answer: true

110) Design patterns are sometimes known as architectural patterns.
TRUE FALSE partially true partially false

Answer: true

111) Design Patterns are groups of objects and their relationships designed to support a â€˜good object designâ€™
TRUE FALSE no idea no option

Answer: true

112) What is â€˜good object design?â€™
yields high cohesion of our objects
has low coupling between our objects
both
none

Answer: both

113) All design involves making decisions
TRUE FALSE no idea no option

Answer: yes

114) Good object design do not involves the assignment of object responsibilities.
TRUE FALSE no idea no option

Answer: false

115) Deciding what methods belong where and how objects interact (their relationships) is
critically important and trivial
critically important and NOT trivial
both
none

Answer: critically important and trivial

116) Patterns that help protect other objects from unanticipated access
immutable and read-only interfaces
immutable and not read-only interfaces
both
none

Answer: immutable and read-only interfaces

117) Patterns where you use delegation to gain access to
Adaptor
FaÃ§ade
Proxy pattern
all

Answer: facade

118) Patterns that assist us in separating concerns
observer singleton iterator facade

Answer: facade

119) A pattern is the outline of a reusable solution to a general problem encountered in a particular context
TRUE FALSE - A

Answer: true

120) A pattern is the outline of a reusable solution to a specific problem encountered in a particular context
TRUE FALSE - B

Answer: true

121) A pattern is the outline of a reusable solution to a specific problem encountered in a general context
TRUE FALSE - B

Answer: true

122) A pattern is the outline of a reusable solution to a general problem encountered in a

general context
TRUE FALSE - B

Answer: true

123) A good pattern should
Be as general as possible
Be as specific as possible
-
A

Answer: Be as general as possible

124) Pattern contain a solution that has been proven to effectively solve the problem in the indicated context.
Good pattern
Not good pattern
General pattern
Not general pattern

Answer: Good pattern

125) Studying patterns is an effective way to learn from ---
the experience of others
the experience of project manager
the experience of the team laeder
the experience of design team only

Answer: the experience of others

126) The general situation in which the pattern applies
contextproblem solution project

Answer: context

127) A short sentence or two raising the main difficulty.
contextproblem solution project

Answer: solution (not sure)

128) The issues or concerns to consider when solving the problem

forces problem solution project

Answer: forces

129) The recommended way to solve the problem in the given context.
contextproblem solution project

Answer: solution

130) Solutions that are inferior or do not work in this context.
Antipatterns Related patterns references domain

Answer: Antipatterns

131) Patterns that are similar to this pattern.
Antipatterns Related patterns co pattern domain

Answer: Related patterns

132) Who developed or inspired the pattern
Antipatterns References Related patterns Solution

Answer: 🥴

133) Creational Patterns
Factory method singleton prototype all

Answer: all

134) Structural Patterns
Adapter
Proxy
Facade
all

Answer: all

135) patterns are a common design vocabulary
allows engineers to abstract a problem and talk about that abstraction in isolation from its implementation
embodies a culture; domain-specific patterns increase design speed
both
none

Answer: both

136) patterns capture design expertise and allow that expertise to be communicated
promotes design reuse and promotes design reuse
promotes design reuse
promotes design reuse
none

Answer: promotes design reuse

137) What are Benefits of using patterns
improve documentation
understandability
both
none

Answer: both

138) Iterator pattern that is
supports concurrent iteration and element removal
uniform interface for traversing many different data structures
an object that provides a standard way to examine all elements of any collection
all

Answer: all

139) Observer pattern is nothing but --
objects whose state can be watched
objects whose instance can be watched
objects whose class can be watched
objects whose interface can be watched

Answer: objects whose state can be watched

140) ----represent solutions to problems that arise when developing software within a particular context
Design software
Design patterns
Design hardware
Analysis patterns

Answer: Design patterns

141) Patterns capture the --- structure and collaboration among key participants in software designs
static
Dynamic
A and B
None of above

Answer: static

142) Patterns facilitate ----- of successful software architectures and designs
Updating
Addition
Manipulate
Reuse

Answer: Reuse

143) Application domain of Design patterns are
CAD and CAE
cellular network management and telecomm switches
program visualization
All of above

Answer: All of above

144) technical areas of Design patterns are
user interface
communications
persistent objects
All of above

Answer: All of above

145) A Design Pattern do not Describes a recurring design structure with
identifies classes
Encapsulation
responsibilities
Collaborations

Answer: identifies classes

146) A Design Pattern Describes a recurring design structure with
applicability
trade-offs
consequences
All of above

Answer: all

147) In Design pattern what is content intent?
objects/classes and their responsibilities
situations where pattern can be applied
Problem and Context
scenario illustrates a design problem

Answer: Problem and Context

148) In the Design pattern what is motivation?
objects/classes and their responsibilities
situations where pattern can be applied
Problem and Context
the scenario illustrates a design problem

Answer: the scenario illustrates a design problem

149) In the Design pattern what is participants?
objects/classes and their responsibilities
situations where pattern can be applied
Problem and Context
scenario illustrates a design problem

Answer: objects/classes and their responsibilities

150) In Design pattern what is Applicability
objects/classes and their responsibilities
situations where pattern can be applied
Problem and Context
the scenario illustrates a design problem

Answer: situations where pattern can be applied

151) In Design pattern what is Structure?
graphical representation of classes
objects/classes and their responsibilities

how participants collaborate
trade-offs and results

Answer: graphical representation of classes

152) In the Design pattern what are Collaborations for complex projects?
graphical representation of classes
objects/classes and their responsibilities
how participants collaborate
trade-offs and results

Answer: how participants collaborate

153) Which of the following is correct about Creational design patterns.
These design patterns are specifically concerned with communication between objects.
These design patterns provide a way to create objects while hiding the creation logic, rather than instantiating objects directly using new opreator.
These design patterns concern class and object composition. Concept of inheritance
None of the above.

Answer: These design patterns concern class and object composition. Concept of inheritance

154) Which of the following pattern is used when we need to decouple an abstraction from its implementation so that the two can vary independently?
Bridge Pattern

Answer: Bridge Pattern

155) In Design pattern what is the Consequences for the life-critical project?
graphical representation of classes
objects/classes and their responsibilities
how participants collaborate
trade-offs and results

Answer: trade-offs and results