

JNTU ONLINE EXAMINATIONS [Mid 2 – OOAD]

1. _____ are the capabilities and conditions to which the system must conform → **Requirements**
2. _____ are the mechanisms to discover and record requirements → **Use Cases**
3. _____ are the primary mechanisms in the Unified Process to describe system behavior → **Use Cases**
4. _____ are the stories of using a system to meet the goals → **Use Cases** vsnr2011@gmail.com
5. _____ are used to model records of activities that describe what happened in the past/what needs to be done later → **Events**
6. _____ captures the intended behavior of a system → **Use Case**
7. _____ defines a promise or contract of how a system will behave → **Use Case section**
8. _____ defines the stakeholders view of the product to be developed → **Vision**
9. _____ diagram illustrates input and output events related to the system under discussion → **Sequence diagram**
10. _____ diagram illustrates use case realizations → **Sequence diagram**
11. _____ diagram is used to model a physical database → **Component Diagram**
12. _____ diagram is used to model an operation → **Activity Diagram** vsnr2011@gmail.com
13. _____ diagram is used to model Client- Server systems → **Deployment Diagram**
14. _____ diagram is used to model physical aspects of an object-oriented system → **Deployment Diagram**
15. _____ diagram is used to model static implementation view of a system → **Component Diagram**
16. _____ diagram is used to model the requirements of a system → **Use Case Diagram**
17. _____ diagram is used to model the vocabulary of a system → **Class diagram**
18. _____ diagram shows the systems topology and distribution of components. → **Deployment Diagram**
19. _____ distinguishes a sequence diagram from a collaboration diagram → **Path**
20. _____ is a collection of operations used to specify a service of a component → **Interface**
21. _____ is a contract or obligation of a class → **Responsibility**
22. _____ is a path from one activity state to the next activity state → **Transition**
23. _____ is an important building block in modeling the physical aspects of a system → **Node**
24. _____ is an instance of an association → **Link**
25. _____ is denotation for the time at which an event occurs → **Timing mark**
26. _____ is used to denote a collection of instances → **Multi object**
27. _____ is used to graphically represent the flow of events of a use case → **sequence diagram**
28. _____ is used to indicate the version number of the source code file vsnr2011@gmail.com → **Tagged Value**
29. _____ is used to model Application Programming Interfaces → **Component**
30. _____ is used to model common families of processes or threads → **Active class**
31. _____ is used to model physical and replaceable part of a system [11S05] → **Component**
32. _____ is used to model processors → **Node**
33. _____ is used to model source code [12S05] vsnr2011@gmail.com → **Component**
34. _____ is used to model the actions that cause components migrate → **Interaction Diagram**
35. _____ is used to model the distribution of components → **Node**
36. _____ is used to organize the nodes → **Package** vsnr2011@gmail.com
37. _____ is used to represent concurrent flows in an Activity Diagram → **Synchronization bar**
38. _____ is used to show which one, out of a set of transitions actually fires on a particular occasion → **Guard Condition**
39. _____ is used to specify new kinds of components → **Stereotype**
40. _____ is used to structure the behavioral things in a model → **Use case**
41. _____ language is used to express constraints in models → **Object constraint language**
42. _____ may have two or more incoming transitions and one outgoing transition → **Join**
43. _____ model visualizes the conceptual classes in the problem domain → **Domain model**
44. _____ provide services to the system under design → **Supporting Actor**
45. _____ relationship between use cases means that the base use case explicitly incorporates the behavior of another use case at a location specified in the base → **Include**
46. _____ relationship exists between the classes Student and Book vsnr2011@gmail.com → **Association**
47. _____ relationship exists between the classes Student and Library → **Aggregation**
48. _____ relationship is NOT present in a Use Case Diagram → **Realization Relationship**
49. _____ relationship is used to model 'Inheritance' property of Object Oriented systems → **Generalization**
50. _____ represents a role that plays within a system → **Actor**
51. _____ state "what must always" be true beginning a scenario in the use case → **Pre-conditions**
52. _____ stereotype is used to specify a component that represents a database table → **Table**
53. _____ uses the services of the system under design to fulfill the goals → **Primary Actor**
54. A sequential state machine may have → **At most one initial state and one final state**
55. A set of objects or components that are allocated to a node as a group is called → **Distribution unit**
56. A state machine whose actions are all attached to states is called → **Moore machine**
57. A state that is nested inside another state is called → **Composite State** vsnr2011@gmail.com
58. A Use case captures _____ of a system → **Behavior**

-
59. A use case is realized by → **A Collaboration** vsnr2011@gmail.com
 60. A use case is rendered as → **An ellipse**
 61. Absolute time of an event is modeled as → **Timing Constraint**
 62. Activity diagram is a special kind of → **State Chart Diagram**
 63. Activity diagram is used to model _ _ _ _ _ aspects of a system → **Dynamic aspects**
 64. Actors are connected to use cases only by → **Association relationship**
 65. Actors similar to one another are organized using _ _ _ _ _ relationship → **Generalization**
 66. An interface that a component realizes is called as vsnr2011@gmail.com → **Export interface**
 67. Animation of a model against the execution of a deployed system is an example of → **Reverse Engineering**
 68. Attribute visibility from object A to object B exists when → **B is an attribute of A**
 69. By default methods are vsnr2011@gmail.com → **Public**
 70. Call Events are handled by → **A Method**
 71. Component Diagram is a special kind of _ _ _ _ _ → **Class Diagram**
 72. Component diagrams are commonly used to model → **Physical database**
 73. Components are organized using → **Generalization**
 74. Components that are essentially the residue of the deployment process are called → **Work Product Components**
 75. Components that are necessary and sufficient to form an executable system are called → **Deployment Components** vsnr2011@gmail.com
 76. Conceptual class consists only → **Attributes**
 77. Consider the example of autonomous robot. Then which of the following is an example of an External Event. → **Collision**
 78. Creating a new system by adding new components and replacing old ones is called → **Binary replaceability**
 79. Deployment Diagram is a special kind of → **Class Diagram** vsnr2011@gmail.com
 80. Deployment diagram is used to model → **The static aspects of a system**
 81. Diagram is used to model reactive objects. → **State Chart Diagram**
 82. Diagram is used to model the distribution of objects → **Object Diagram**
 83. Diagram is used to show the state space of a given class. → **State Chart Diagram**
 84. Events that are not handled in that state but are postponed and queued for handling by the object in another state are called as → **Deferred Events**
 85. Events that can cause transitions to fire are known as → **Triggers**
 86. Executable atomic computations are called as → **Actions**
 87. Executable atomic computations are called as → **Action States**
 88. Executable non atomic computations are called as → **Activity States**
 89. Extend relationship is rendered using _ _ _ _ _ → **Dependency relationship**
 90. Extension scenarios are also called as → **Alternative Flows**
 91. Forward Engineering is possible for an Activity Diagram especially if the context of the diagram is _ _ _ → **An Operation**
 92. Forward Engineering is the process of translating → **Model to Code** vsnr2011@gmail.com
 93. Forward engineering of a collaboration diagram is possible especially if the context of the diagram is → **An operation**
 94. Forward Engineering of a state chart diagram is possible especially if the context of the diagram is → **A Class**
 95. Graphically a node is rendered as → **Cube**
 96. Graphically an active class is rendered as a rectangle with → **Thick Lines**
 97. Graphically, a component is rendered as → **A Rectangle with tabs**
 98. Graphically, a timing constraint is rendered as → **A Constraint**
 99. Graphically, location is rendered as → **A tagged value** vsnr2011@gmail.com
 100. In a Use Case Diagram, use cases are adorned with _ _ _ _ _ to assert nonfunctional requirements → **Notes**
 101. In an Activity Diagram, organizing the activities into groups is called vsnr2011@gmail.com → **Swim lane**
 102. In an Activity Diagram, transitions belongs to → **Completion Transitions**
 103. In class box attributes are shown in → **Second compartment**
 104. In Library system, the multiplicity of the association between student and book corresponds to → **One-to-many**
 105. In practice node names are → **Short nouns**
 106. In practice, component names are → **Short nouns**
 107. In the UML, the system as a whole can be represented by a → **Class** vsnr2011@gmail.com
 108. In UML _ _ _ _ _ keyword is used to model time events → **After**
 109. In UML _ _ _ _ _ is used to model each independent flow of control → **Active Object**

110. In UML Exceptions are modeled as → **Stereo Typed classes** vsnr2011@gmail.com
111. In UML signals are modeled as → **Stereo Typed classes**
112. In UML, a state is graphically rendered as → **Rounded rectangle**
113. In UML, activity state is graphically rendered as → **A lozenge shape**
114. In UML, asynchronous message is rendered as _ _ _ _ → **Half Arrow**
115. In UML, physical things are modeled as → **Components** vsnr2011@gmail.com
116. In UML, processes are modeled as → **Stereotyped Active Classes**
117. In UML, transition is graphically rendered as → **A Directed line**
118. In Use Case Diagrams, common behavior of actors are generally named as → **Use Cases**
119. Internal event occurs when → **A method is invoked via a message**
120. Metadata is stored in → **Data dictionary**
121. Multiplicity ``1..*`` represents → **One or more** vsnr2011@gmail.com
122. Navigability in the association implies → **Visibility**
123. Non software things are modeled using → **Classes**
124. Non-functional requirements of Library System are captured in → **Supplementary Specification**
125. Object -oriented analysis emphasizes on → **Finding and describing the objects**
126. Objects placed in an Activity Diagram are connected to the activity or transition using _ _ _ relationship → **Dependency**
127. On a domain model with n different conceptual classes, there can be → **n(n-1) associations**
128. Pre-conditions of an initial state and post-conditions of a final state of the workflow are important in modeling _ _ _ _ → **Boundaries of the workflow**
129. Process view primarily addresses → **Performance**
130. Realization of a use case is specified by _ _ _ _ _ → **A collaboration**
131. Sequence diagram is used to model → **Flows of control by time ordering**
132. Sequence diagrams are part of → **Use Case model** vsnr2011@gmail.com
133. State chart Diagrams are needed → **When a class has complex life cycle**
134. State independent means that the object responds → **The same way to a different event**
135. State that is active after the completion of the transition is called → **Target State**
136. The allocation of an object to a new node is rendered as → **a ``copy`` stereotyped message**
137. The behavior of a system is modeled using _ _ _ _ _ → **Use Case Diagram**
138. The behavior of a use case is specified by _ _ _ _ _ → **Flow of events**
139. The behavior of an active class is specified by → **A State Machine**
140. The collaboration of active and passive objects is specified Using → **Component Diagrams**
141. The deployment view of a system is modeled using → **Development Diagram**
142. The domain model of Library System consists → **Book class with attributes title, price**
143. The elements of an interface are → **Operations**
144. The idea of use cases to describe the functional requirements was introduced by → **Ivar Jacobson**
145. The name of a system event starts with → **A Verb** vsnr2011@gmail.com
146. The names of use cases are generally given as → **Verb phrases**
147. The path name of a component is the name → **Prefixed by the name of the package in which it lives**
148. The relationship between a node and component can be shown explicitly using → **Dependency relationship**
149. The Relationship between two states is called _ _ _ _ _ → **Transition**
150. The relationship between two states is called → **Transition**
151. The scenario of a use case is graphically represented using → **Sequence diagram**
152. The set of significant decisions about the organization of a software system is called → **Software architecture**
153. The static deployment view of a system primarily addresses → **Installation**
154. The terms and definitions of library system are captured in → **Glossary** vsnr2011@gmail.com
155. Time critical systems are called as → **Real time Systems**
156. Transitions that are handled without causing a change in state are called → **Internal Transitions**
157. Use Case Diagrams are used to model _ _ _ _ _ view of a system → **Use Case View**
158. Using UML, a domain model is illustrated with a set of → **Class diagrams with no operations**
159. Which of the following class present in the class diagram of Library System? → **Student**
160. Which of the following conceptual class presents in the domain model of Library System? → **Register**
161. Which of the following diagram is used to model adaptable systems → **Component Diagram**
162. Which of the following diagram is used to model business workflows? → **Activity Diagram**
163. Which of the following diagram is used to model embedded systems → **Deployment Diagram**
164. Which of the following diagram is used to model object Migration → **Collaboration Diagram**
165. Which of the following diagram is used to model the dynamic aspects of a system? → **State Chart Diagram**
166. Which of the following diagram is used to represent structural organization of objects → **Collaboration diagram** vsnr2011@gmail.com

167. Which of the following diagram view the whole system as a block box? → **Use Case Diagram**
168. Which of the following element is NOT present in a deployment diagram? → **Tagged value**
169. Which of the following evaluates to an absolute value of Time? → **Timing Expression**
170. Which of the following is a change event → **When (11:30AM)** vsnr2011@gmail.com
171. Which of the following is a functional requirement? → **Behavior**
172. Which of the following is a quality attribute? → **Usability**
173. Which of the following is a strategy for identifying conceptual classes? → **Identifying noun phrases**
174. Which of the following is a valid actor for credit card validation system? → **Retail Institution**
175. Which of the following is a valid actor for library system? → **Student**
176. Which of the following is a valid use case for library system? → **Issue Book** vsnr2011@gmail.com
177. Which of the following is an Asynchronous Event → **Signal Event**
178. Which of the following is an example of execution component → **COM+ Object**
179. Which of the following is NOT present in a Component Diagram? → **Class**
180. Which of the following is not present in a sequence diagram? → **Sequence number**
181. Which of the following is not present in a State Chart Diagram? → **Fork**
182. Which of the following is NOT present in a Use Case Diagram? vsnr2011@gmail.com → **Collaboration**
183. Which of the following is NOT present in an Activity Diagram? → **Events**
184. Which of the following is present in a nested concurrent state machine → **Concurrent sub state**
185. Which of the following is true regarding a process → **It represents an independent flow of control**
186. Which of the following is used to bridge logical and physical models → **Interface**
187. Which of the following is used to model the life time of an object → **State Machine**
188. Which of the following is used to model the seams in a system → **Interface** vsnr2011@gmail.com
189. Which of the following is used to organize the use cases → **Generalization**
190. Which of the following is wrong with respect to a thread → **Threads are nested inside another thread**
191. Which of the following is WRONG with respect to components and classes? → **Both represent physical things**
192. Which of the following process model is best suited for developing object-oriented systems? → **Unified Process model**
193. Which of the following relationship is NOT present between nodes → **Realization**
194. Which of the following relationship is not present in a deployment diagram? → **Generalization**
195. Which of the following represents a computational resource? → **Node** vsnr2011@gmail.com
196. Which of the following represents physical packaging of logical elements → **Component**
197. Which of the following shows the set of components and their relationship → **Component Diagram**
198. Which of the following stereotype is used to specify a component that represents a source code? → **File**
199. While modeling an operation, use forking and joining only if the operation is owned by → **An Active Class** vsnr2011@gmail.com



ALL THE BEST FOR **FINAL EXAMS-BY**

NOBEL EXCELLERS