



# INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

## INFORMATION TECHNOLOGY

### DEFINITIONS AND TERMINOLOGY QUESTION BANK

Course Name	:	<b>OBJECT ORIENTED ANALYSIS AND DESIGN</b>
Course Code	:	<b>ACSB10</b>
Program	:	<b>B.Tech</b>
Semester	:	<b>FIVE</b>
Branch	:	<b>INFORMATION TECHNOLOGY</b>
Section	:	<b>A &amp; B</b>
Academic Year	:	<b>2020 - 2021</b>
Course Faculty	:	<b>Dr. C Raghavendra, Associate Professor Dr. M Mohana Roopa, Professor Mr. G Chandra Sekhar, Assistant Professor Mr. R M Noorullah, Associate Professor Ms. N Shalini, Assistant Professor</b>

#### COURSE OBJECTIVES:

<b>The course should enable the students to:</b>	
I	The basic and advanced building blocks of Unified Modeling Language for analysis and design of software systems.
II	The Object-oriented approach for analysis and design of System/Subsystem/Functional units based on the given specifications through UML Diagrams.
III	The implementation of design document of real time software applications using advanced CASE tools

### DEFINITIONS AND TERMINOLOGY QUESTION BANK

S. No	Question	Answer	Blooms Level	CO
<b>UNIT - I</b>				
1	Define Model?	A model is a simplification of reality.	Remember	CO 1
2	What is Analysis and Design?	<b>Analysis:</b> Basically, it is the process of determining what needs to be done before how it should be done. In order to accomplish this, the developer refers the existing systems and documents. So, simply it is an art of discovery.	Remember	CO 1
3	What is meant by Design?	<b>Design:</b> It is the process of adopting/choosing the one among the many, which best accomplishes the users needs. So, simply, it is compromising mechanism.	Remember	CO 1

4	What are the steps involved in designing?	<p>Before getting into the design the designer should go through the SRS prepared by the System Analyst.</p> <ul style="list-style-type: none"> <li>○ The main tasks of design are Architectural Design and Detailed Design.</li> <li>○ In Architectural Design we find what the main modules in the problem domain are.</li> <li>○ In Detailed Design we find what should be done within each module.</li> </ul>	Remember	CO 1
5	What are the main underlying concepts of object orientation?	Objects, messages, class, inheritance and polymorphism are the main concepts of object orientation.	Remember	CO 1
6	Define persistent objects?	Persistent refers to an object's ability to transcend time or space. A persistent object stores/saves its state in a permanent storage system without losing the information represented by the object.	Remember	CO 1
7	What are non-persistent objects?	A non-persistent object is said to be transient or ephemeral. By default objects are considered as non-persistent.	Remember	CO 1
8	What are models And meta models?	<p><b>Model:</b> It is a complete description of something (i.e. system).</p> <p><b>Meta model:</b> It describes the model elements, syntax and semantics of the notation that allows their manipulation.</p>	Remember	CO 1
9	What is UML?	The Unified Modeling Language is a visual language for specifying, constructing and documenting the artifacts of systems.	Remember	CO 1
10	What is Inception?	Inception is the initial short step to establish a common vision and basic scope for the Project.	Remember	CO 2
11	What artifacts may start in Inception?	Some sample artifacts are Vision and Business Case, Use-Case Model, Supplementary Specification, Glossary, Risk List & Risk Management Plan, Prototypes and proof-of concepts etc.	Remember	CO 2
12	What is Object-Oriented Design?	During object-oriented design (or simply, object design) there is an emphasis on defining software objects and how they collaborate to fulfill the requirements.	Remember	CO 1
13	What the main advantages of object oriented development?	<ul style="list-style-type: none"> <li>• High level of abstraction</li> <li>• Seamless transition among different phases of software development</li> <li>• Encouragement of good programming techniques</li> <li>• Promotion of reusability</li> </ul>	Remember	CO 1
14	What is object oriented system development methodology?	Object oriented system development methodology is a way to develop software by building self-contained modules or objects that can be easily replaced, modified and reused.	Remember	CO 1
15	Define class diagram.	The main static structure analysis diagram for the system, it represents the class structure of a system including the relationships between class and inheritance structure.	Remember	CO 2

16	Define activity diagram.	A variation or special case of a state machine in which the states are activities representing the performance of operations and the transitions are triggered by the completion of the operations.	Remember	CO 2
17	What is interaction diagram? Mention its types.	Interaction diagrams are diagrams that describe how groups of objects collaborate to get the job done. Interaction diagrams capture the behavior of the single use case, showing the pattern of interaction among objects. <b>Types:</b> Sequence diagram, Collaboration diagram	Remember	CO 2
18	What is sequence diagram?	It is an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment	Remember	CO 2
19	What is collaboration diagram?	It represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects with in collaboration to achieve a desired outcome.	Remember	CO 2
20	Define state chart diagram.	It shows a sequence of states that an object goes through during its life in response to events. A state is represented as a round box, which may contain one or more compartments. The compartments are all optional.	Remember	CO 2
21	Define component diagram.	A component diagram shows the organization and dependencies among a set of components. A component diagram is used to model the static implementation view of a system. This involves modeling the physical thing that reside on a mode, such as executable, libraries, files and documents.	Remember	CO 2
22	Define deployment diagram	Deployment diagram shows the configuration of run time processing elements and the software components, processes and objects that live in them.	Remember	CO 2
23	What is UP?	A software development process describes an approach to building, deploying and possibly maintaining software. The unified process has emerged as a popular iterative software development process for building object oriented system.	Remember	CO 2
24	What is iteration?	A key practice in both the UP and most other modern methods is iterative development. In this lifecycle approach, development is organized into a series of short, fixed length mini projects called iterations	Remember	CO 2
25	What are the phases of unified process?	<ul style="list-style-type: none"> <li>• Inception</li> <li>• Elaboration</li> <li>• Construction</li> <li>• Transition</li> </ul>	Remember	CO 2
26	What is Composition?	Composition, also known as composite aggregation, is a strong kind of whole-part aggregation and is useful to show in some models.	Remember	CO 2
27	What is UML activity diagram?	A UML activity diagram shows sequential and parallel activities in a process, they are useful for modeling business processes, workflows, data flows and complex algorithms.	Remember	CO 2

28	What is Generalization?	Generalization is the activity of identifying commonality among concepts and defining superclass (general concept) and subclass (specialized concept) relationships.	Remember	CO 2
29	What is Aggregation?	Aggregation is a vague kind of association in the UML that loosely suggests whole-part relationships (as do many ordinary associations).	Remember	CO 2
30	What is message and method?	<b>Message:</b> Message essentially are non-specific function calls <b>Method:</b> Methods are similar to functions, procedures or subroutines in more traditional programming languages	Remember	CO 2
<b>UNIT-II</b>				
1	Define Encapsulation?	Encapsulation is property in which data is accumulated and restricted to a particular area only.	Remember	CO 4
2	Define Interface?	Interface defines service to each class.	Remember	CO 4
3	Expand the term OMT	Object Modeling Techniques	Understand	CO 3
4	Define Use case Driven?	It refers to the Utilization of use cases as an effective element for narrating the behavior of any system.	Remember	CO 3
5	Expand the term CASE	Computer Aided Software Engineering	Understand	CO 3
6	Define Forward Engineering.	Building code from Model is known as Forward Engineering	Remember	CO 3
7	Define Reverse Engineering.	Building Model from Code is known as Reverse Engineering	Remember	CO 3
8	What is Design View?	It consists of Classes interfaces and Collaborations that forms the actual definition of a Problem and its solution.	Remember	CO 3
11	Define Class?	Class is a collection of objects that share common attributes and operations.	Remember	CO 4
12	Which diagrams are used to represent Design view of UML architecture?	Class and object diagrams are used to represent Design view of UML architecture.	Remember	CO 4
13	What are constraints of object diagrams?	New, destroy and transient are constraints of object diagrams.	Remember	CO 4
14	What is Class?	A Class is a blueprint that is used to create Object. The Class defines what object can do.	Remember	CO 4
15	What is Class Diagram?	CLASS DIAGRAM gives an overview of a software system by displaying classes, attributes, operations, and their relationships. This Diagram includes the class name, attributes, and operation in separate designated compartments.	Remember	CO 3
16	What is artifact?	An artifact is a classifier that represents some physical entity, piece of information that is used or is produced by a software development process, or by deployment and operation of a system	Remember	CO 2
17	What is Node?	A Node is a deployment target which represents computational resource upon which artifacts may be deployed for execution.	Remember	CO 2

18	What is Device?	A device is a subclass of node which represents a physical computational resource with processing capability upon which artifacts may be deployed for execution.	Remember	CO 2
19	What is Deployment specification?	A deployment specification is an artifact that specifies a set of deployment properties that determine execution parameters of a component artifact that is deployed on a node.	Remember	CO 2
20	What are Artifact Manifestation	Manifestation is an abstraction relationship which represents the concrete physical rendering of one or more model elements by an artifact or utilization of the model elements in the construction or generation of the artifact.	Remember	CO 2
<b>UNIT-III</b>				
1	Which diagram emphasize on structural organization?	Collaboration diagram	Remember	CO 7
2	Define Use case?	Use case describes a set of actions that are to be performed by a system.	Remember	CO 7
3	What are common modeling techniques of use cases?	To model context of a system, To model requirements of a system	Remember	CO 7
4	What are branches in activity diagram?	Branches are a notational convenience, semantically equivalent to multiple transitions with guards	Remember	CO 7
5	Define Fork in activity diagram.	A fork may have one incoming transitions and two or more outgoing transitions	Remember	CO 7
6	Define action state.	Action states are atomic and cannot be decomposed	Remember	CO 8
7	Which diagrams are Isomorphs in nature?	Interaction diagrams	Remember	CO 7
8	Name interaction diagrams	Sequence and collaboration diagrams	Remember	CO 7
9	Which diagrams are used to represent use case view of UML architecture?	Use Case Diagrams are used to represent use case view of UML architecture.	Remember	CO 2
10	Which diagrams are used to represent component view of UML architecture?	Component Diagrams are used to represent component view of UML architecture.	Remember	CO 2
11	Which diagrams are used to represent Deployment view of UML architecture?	Deployment Diagram are used to represent Deployment view of UML architecture.	Remember	CO 2
12	What are Actors?	An actor is something with behavior, such as a person (identified by role), computer system, or organization; for example, a cashier.	Remember	CO 7
13	What are three kinds of Actors?	Primary actor, Supporting actor, offstage actor.	Remember	CO 7
14	What tests can help find useful use Cases?	1. The Boss Test 2. The EBP Test 3. The Size Test	Remember	CO 7
15	Which diagrams are used to represent Logical view of UML architecture?	Interaction and activity diagrams are used to represent Logical view of UML architecture.	Remember	CO 2
16	What is an Activity Diagram?	An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram.	Remember	CO 7

17	What are key parts of sequence diagram	Participant and messages	Remember	CO 7
18	Define scenario.	A Scenario is a specific sequence of actions that illustrates behavior	Remember	CO 7
19	What is an time event?	This refers to an event that stops the flow for a time; an hourglass depicts it.	Remember	CO 8
20	Define Merge Event	A merge event brings together multiple flows that are not concurrent.	Remember	CO 8
21	What is Interrupting Edge	An event, such as a cancellation, that interrupts the flow denoted with a lightning bolt.	Remember	CO 8
22	Define Swimlanes.	Swimlanes group related activities into one column.	Remember	CO 7
23	What are the Benefits of activity diagrams?	<ul style="list-style-type: none"> <li>• Demonstrate the logic of an algorithm.</li> <li>• Describe the steps performed in a UML use case.</li> <li>• Illustrate a business process or workflow between users and the system.</li> </ul>	Remember	CO 7
24	Define the basic components of an activity diagram.	<ul style="list-style-type: none"> <li>• Basic components of activity diagrams are: Action, Decision node, Control flows, Start node and End node</li> </ul>	Remember	CO 7
25	What decision symbol in activity diagram represents?	This symbol represents the branching or merging of various flows with the symbol acting as a frame or container.	Remember	CO 7
26	Define the importance of Activity symbol in activity diagram.	It indicates the activities that make up a modeled process. These symbols, which include short descriptions within the shape, are the main building blocks of an activity diagram.	Remember	CO 7
<b>UNIT-IV</b>				
1	Define call Events?	Call event is the receipt of a request to invoke an operation.	Remember	CO 8
2	Define source state?	It is the state affected by the Transition.	Remember	CO 8
3	Define signal event.	A signal event represents a named object that is dispatched (thrown) asynchronously by one object and then received (caught) by another.	Remember	CO 8
4	Define Signal?	A signal signifies named objects which are transmitted or received among several objects.	Remember	CO 8
5	Define Abstract Product?	Declares an interface for a type of product object	Remember	CO 8
6	What is transition?	A transition is said to have occurred when an object moves from one state to another.	Remember	CO 8
7	What is activity?	Activity refers to the actions that are performed when an object is in particular state.	Remember	CO 7
8	Define event trigger.	An event is the specification of a significant occurrence that has a location in time and space. In the context of state machines, an event is an occurrence of a stimulus that can trigger a state transition.	Remember	CO 8
9	Define component.	A component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces. Graphically, a component is rendered as a rectangle with tabs.	Remember	CO 10

10	Define node.	A node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and, often, processing capability. Graphically, a node is rendered as a cube.	Remember	CO 10
11	Define Action.	An action is an executable atomic computation. Actions may include operation calls the creation or destruction of another object, or the sending of a signal to an object. An action is atomic, meaning that it cannot be interrupted by an event and therefore runs to completion.	Remember	CO 8
12	How signals are modeled?	As stereo typed classes	Remember	CO 8
13	What are contents of state chart diagram?	state chart diagrams contain branches, forks, joins, action states, activity states, objects, initial states, final states, history states	Remember	CO 8
14	What are common modeling techniques of state chart diagrams?	To model a reactive object.	Remember	CO 9
15	Define Component.	A component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces.	Remember	CO 10
16	What are Domain Objects?	Software objects representing domain concepts	Remember	CO 10
17	What is POS system?	A POS system is a computerized application used (in part) to record sales and handle payments; it is typically used in a retail store	Remember	CO 12
18	What hardware are used in POS?	Computer and Barcode Scanner	Remember	CO 12
19	What are service applications in POS?	Third Party Ytax calculator and Inventory Control	Remember	CO 12
20	What are the goals of cashier in POS?	process sales, handle returns, cash in, cash out	Remember	CO 12
21	What are the goals of system administrator in POS?	manage users, manage security, manage system tables	Remember	CO 12
22	What are the goals of manager in POS?	start up, shut down	Remember	CO 12
23	What are the goals of sales activity system in POS?	analyze sales data	Remember	CO 12

#### UNIT-V

1	Define Active class?	Active class possesses objects which consist of processes and threads.	Remember	CO 5
2	Define Component?	Components bind several classes, interfaces and collaborations by physical representation.	Remember	CO 10
3	Describe Node?	Node is an element which holds certain locations in the memory sometimes along with processing capabilities.	Remember	CO 10
4	What is interface?	An interface (small circle or semi-circle on a stick) describes a group of operations used (required) or created (provided) by components. A full circle represents an interface created or provided by the component. A semi-circle represents a required interface, like a person's input.	Remember	CO 2

5	What is Port?	Ports are represented using a square along the edge of the system or a component. A port is often used to help expose required and provided interfaces of a component	Remember	CO 10
6	Define Component Diagram?	A Component diagrams shows the organization and dependencies among a set of components. A component diagrams are used to model the static implementation view of a system.	Remember	CO 10
7	Define Deployment diagram.	Deployment Diagram shows the configuration of run-time processing elements and the software components, processes, and objects that live in them.	Remember	CO 10
8	What is the basic element of a deployment diagram? And give its types.	The basic element of a deployment diagram is <b>node</b> . Types are: 1. Device node 2. Execution environment node (EEN)	Remember	CO 10
9	Define Component.	A component represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment.	Remember	CO 10
10	What is the use of Deployment diagram?	Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.	Remember	CO 10
11	What Deployment diagram consists of?	Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.	Remember	CO 10
12	Define the purpose of Deployment diagram.	The purpose of Deployment diagram is: <ul style="list-style-type: none"> <li>• Visualize the hardware topology of a system.</li> <li>• Describe the hardware components used to deploy software components.</li> <li>• Describe the runtime processing nodes.</li> </ul>	Remember	CO 10
13	What type of parameters the deployment diagram controls?	An efficient deployment diagram is very important as it controls the following parameters – <ul style="list-style-type: none"> <li>• Performance</li> <li>• Scalability</li> <li>• Maintainability</li> <li>• Portability</li> </ul>	Remember	CO 10
14	Where to Use Deployment diagrams?	Deployment diagrams are mainly used by system engineers. These diagrams are used to describe the physical components (hardware), their distribution, and association.	Remember	CO 10
15	What are uses of Deployment diagrams?	Usage of deployment diagrams can be described as follows: <ul style="list-style-type: none"> <li>• To model the hardware topology of a system.</li> <li>• To model embedded system.</li> <li>• To model hardware details for a client/server system.</li> <li>• To model hardware details of a distributed application.</li> <li>• Forward and reverse engineering.</li> </ul>	Remember	CO 10
16	Where to use Component diagrams?	Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.	Remember	CO 10



17	Define the benefits of component diagram.	The benefits of component diagram are: <ul style="list-style-type: none"> <li>• Imagine the system's physical structure.</li> <li>• Pay attention to the system's components and how they relate.</li> <li>• Emphasize the service behavior as it relates to the interface.</li> </ul>	Remember	CO 10
17	What is Connector?	A Connector is a relation between ports of components. If one port provides interface required by the other port, they can be linked together.	Remember	CO 2
18	Define the structure of a component.	A component is represented with classifier rectangle stereotypes as << component >>. Component details are hidden for the outside world. The name of a component is placed at the center of a rectangle.	Remember	CO 10
19	What are Interfaces in component diagram?	The interface is a named set of public features. It separates the specification of functionality from its implementation by a class diagram or a subsystem.	Remember	CO 10
20	What are the advantages of interfaces?	Advantages of interfaces are: <ul style="list-style-type: none"> <li>• It increases the flexibility and extensibility of a class.</li> <li>• It decreases the implementation dependencies.</li> </ul>	Remember	CO 2
21	What are the disadvantages of interfaces?	Disadvantages of interfaces are: <ul style="list-style-type: none"> <li>• Extra flexibility leads to complex classes.</li> <li>• Too many interfaces make systems hard to understand.</li> </ul>	Remember	CO 2
22	What are Subsystems?	It is a logical construct which is used to break down an extensive system into smaller systems which are known as subsystems.	Remember	CO 2
23	What is the use of Component diagram?	Component diagrams are used to describe the working and behavior of various components of a system.	Remember	CO 10

Signature of the Faculty

HOD, IT