MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu.

Estd. 2000

MUST KNOW CONCEPTS	MKC

MCA	2021-2022

Course Code & Course Name   21CAB12   **Object Oriented Analysis & Design**

Year/Sem/Sec   **I / II / -**

S.No	Term	Notation (Symbol)	Concept/Definition/Meaning/Units/Equation/ Expression	Units
			Unit-I : Introduction	
1	Object-Oriented Analysis and Design	OOAD	OOAD is a technical approach used in the analysis and design of an application or system through the application of the object-oriented paradigm and concepts including visual modeling.	I
2	Object	--	It is a runtime instance of a Class	I
3	Method	--	A method is an action which an object is able to perform.	I
4	Message	--	A Message is a named element that defines one specific kind of communication between lifelines of an interaction.	I
5	Encapsulation	--	It means to bind together code and the data it uses. It is the basis of object-oriented programming.	I
6	Polymorphism	--	The ability of Objects belonging to different Classes to respond to Operations of the same name, each one according to the right Class-specific behavior	I
7	Inheritance	--	It defines where a new more specific Class derives part of its definition from an existing more general Class	I
8	Static Binding	--	The binding which can be resolved at compile time by compiler is known as static or early binding.	I
9	Dynamic Binding	--	A binding in which name can be associated with the class during execution time	I
10	Aggregation	--	It is a special type of association used to represent a stronger relationship between two classes than a regular association	I
11	Association	--	An association is a structural relationship that specifies that the objects of one thing are connected with the objects of another	I

12	Generalization	--	A relationship between a specific classifier (typically a class) to a more general classifier asserting that the general classifier contains common features among both the specific classifier and the general classifier.	I
13	Classes	--	A class represents a collection of objects having same characteristic properties that exhibit common behavior.	I
14	Information Hiding	--	It is the process of hiding the details of an object or function.	I
15	Class Hierarchy	--	It defines the hierarchical relationship in inheritance, whereby from a super-class, a number of subclasses may be derived which may again have subclasses and so on.	I
16	Composition	--	A specific type of relationship describing how one Object is composed of another Object	I
17	Identity	--	It describes the property of objects that distinguishes them from other objects.	I
18	Persistence	--	An object occupies a memory space and exists for a particular period of time.	I
19	Meta classes	--	A Meta classes is one whose instances are also classes	I
20	Prototyping	--	It is a foundational process for developing a new product through the physical representation of an idea.	I
21	Component Based Development	CBD	It is a procedure that accentuates the design and development of computer-based systems with the help of reusable software components.	I
22	Computer-Aided Software Engineering	CASE	It is the domain of software tools used to design and implement applications.	I
23	Rapid Application Development	RAD	It is based on prototyping and iterative model with no (or less) specific planning.	I
24	Software Development Life Cycle	SDLC	It is a process used by the software industry to design, develop and test high quality software's.	I
25	Incremental Testing	--	It is the process of verifying the interfaces and interaction between modules.	I
Unit-II :Methodology and UML				
26	Object Modeling Techniques	OMT	It is an methodology of object oriented analysis, design and implementation that focuses on creating a model of objects from the real world and then to use this model to develop object–oriented software.	II
27	Object-Oriented Software Engineering	OOSE	A software design technique that is used in software design in object-oriented programming. It includes a requirement, an analysis, a design, an implementation and a testing model.	II

28	Unified Modeling Language	UML	It is a graphical language for OOAD that gives a standard way to write a software system's blueprint.	II
29	Class Diagram	--	It is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.	II
30	Object Diagram	--	A diagram that shows a complete or partial view of the structure of a modeled system at a specific time.	II
31	Component Diagram	--	Describes the organization and wiring of the physical components in a system.	II
32	Deployment Diagram	--	Diagram used to visualize the topology of the physical components of a system, where the software components are deployed.	II
33	Use case Diagram	--	A graphical depiction of a user's possible interactions with a system.	II
34	Interaction Diagram	--	Interaction diagram is a type of UML diagram that's used to capture the interactive behavior of a system.	II
35	State-Chart Diagram	--	A state diagram is used to represent the condition of the system or part of the system at finite instances of time.	II
36	Activity Diagram	--	Visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram.	II
37	Qualifier	--	A qualifier is a value that selects a unique object from the set of related objects across an association.	II
38	Multiplicity	--	A specification of the number of possible occurrences of a property, or the number of allowable elements that may participate in a given relationship.	II
39	Reification	--	It is the process by which an abstract idea about a computer program is turned into an explicit data model or other object created in a programming language.	II
40	System Boundary Boxes	--	A box that sets a system scope to use cases.	II
41	Packages	--	A package is a collection or grouping of related classes or of classes with related functionality.	II
42	Actors	--	A role that a user takes when invoking a use case.	II
43	Extensibility	--	It is a measure of the ability to extend a system and the level of effort required to implement the extension.	II
44	Object Lifeline	--	An object is shown as a vertical dashed line called a lifeline. Lifeline represents the existence of object at a particular time.	II
45	Object Constraint Language	OCL	A declarative language for describing rules and constraints that apply to UML models	II
46	Role	--	Description of the part played in an Association by one of the Classes in the Association	II
47	Swimlane	--	A swimlane (as in swimlane diagram) is used in process flow diagrams, or flowcharts, that visually distinguishes job sharing and responsibilities for	II

			sub-processes of a business process.	
48	System model	--	The logical UML model being represented through one or more UML diagrams	II
49	Transition	--	A movement from one State to another in a State machine diagram	II
50	Note	--	A note is a symbol for rendering constraints and comments attached to an element or collection of elements.	II
Unit-III : Object Oriented Analysis				
51	Abstraction	--	The process of picking out common features and deriving essential characteristics from objects and procedure entities that distinguish it from other kinds of entities.	III
52	Attribute	--	An attribute is a specification that defines a property of an object, element, or file.	III
53	Analysis	--	Analysis emphasizes an investigation of the problem and requirements, rather than a solution.	III
54	Design	--	It is the process by which an agent creates a specification of a software artifact intended to accomplish goals, using a set of primitive components and subject to constraints	III
55	Coding	--	It is the process of transforming the design of a system into a computer language format.	III
56	Testing	--	It is the process of evaluating and verifying that a software product or application does what it is supposed to do.	III
57	Deployment	--	It refers to the process of running an application on a server or device.	III
58	Business Object Analysis	--	The traditional approaches to software development identified and defined rigid flows of procedures to support business processes or activities.	III
59	Uses Association	--	It defines a relationship between classes of objects that allows one object instance to cause another to perform an action on its behalf.	III
60	Extends Association	--	It represents a directed relationship that specifies how and when the behavior defined in usually supplementary extending use case can be inserted into the behavior defined in the extended use case	III
61	User	--	User is a stakeholder who will interact with a completed system either directly or indirectly	III
62	80-20 Rule	--	The 80-20 rule, also known as the Pareto Principle, is an aphorism which asserts that 80% of outcomes result from 20% of all causes for any given event.	III
63	Classical	--	A classification sorts objects into rigid, clearly	III

	Classification		defined categories based on rules.	
64	Fuzzy Classes	--	It define the transformation or remap of the input values to new values based on a specified function.	III
65	Relevant Classes	--	Each class must have a purpose and every class should be clearly defined and necessary.	III
66	Irrelevant Classes	--	These classes have no purpose and are unnecessary.	III
67	Common Class Pattern	--	Based on knowledge base of the common classes , proposed by researchers like shlaer and mellor Patterns for finding class and object	III
68	Dependency	--	A relationship between two or more objects in which an object depends on the other object or objects for its implementation.	III
69	Multiple Inheritance	--	It is a feature of some object-oriented computer programming languages in which an object or class can inherit characteristics and features from more than one parent object or parent class.	III
70	Events	--	An event is the specification of a significant occurrence that has a location in time and space.	III
71	Ternary Association	--	A ternary relationship is an association among three entities. This type of relationship is required when binary relationships are not sufficient to accurately describe the semantics of the association.	III
72	Derived Association	--	Derived associations and derived attributes can be calculated from other associations and attributes, respectively, on a class diagram.	III
73	Noun Phrase Approach	--	Nouns in the textual description are considered to be classes and verbs to be methods of the classes.	III
74	Documentation	--	A documentation is written text or illustration that accompanies computer software or is embedded in the source code.	III
75	Object Oriented Analysis	--	Object Oriented Analysis (OOA) is process of discovery where a development team understands and models the requirements of the system.	III
Unit-IV : Object Oriented Design				
76	Benchmarking	--	Benchmarking is the practice of comparing business processes and performance metrics to industry bests and best practices from other companies.	IV
77	Corollary	--	A corollary is a proposition that follows from an axiom or another proposition that has. been proven	IV
78	Axiom	--	An axiom is a fundamental truth that always is observed to be valid and for which there is no counterexample or exception.	IV
79	Access Layer	--	It is a layer of a computer program which provides simplified access to data stored in persistent storage of some kind, such as an entity-relational database.	IV
80	View Layer	--	It is the only exposed objects of an application with which users can interact.	IV

81	Coupling	--	Coupling is the degree of interdependence between software modules; a measure of how closely connected two routines or modules are; the strength of the relationships between modules.	IV
82	Cohesion	--	Cohesion refers to the degree to which the elements inside a module belong together	IV
83	Protocols	--	A set of rules or procedures for transmitting data between electronic devices, such as computers.	IV
84	Class Visibility	--	Class visibility defines whether attributes and operations of specific classes can be seen and used by other classes.	IV
85	Private Scope	--	A private member is visible only from within the class.	IV
86	Public Scope	--	A public member is visible from anywhere in the system.	IV
87	Protected Scope	--	A protected member is visible from within the class and from the subclasses inherited from this class, but not from outside.	IV
88	Constructor	--	Constructor can be defined as a special member function which is used to initialize the objects of the class with initial values.	IV
89	Destructor	--	A destructor is a special method called automatically during the destruction of an object.	IV
90	Stereotype	--	Stereotypes is extensibility mechanisms in UML which allows designers to extend the vocabulary of UML in order to create new model elements.	IV
91	Object Storage	--	Object storage combines the pieces of data that make up a file, adds all its relevant metadata to that file, and attaches a custom identifier.	IV
92	Object Interoperability	--	It refers to the basic ability of different computerized products or systems to readily connect and exchange information with one another,	IV
93	Transient Data	--	Data that are created within an application session. At the end of the session, the data are discarded or reset back to their default state.	IV
94	Persistent Data	--	Persistent means that the object has been saved to the database	IV
95	Database Management System	DBMS	Database Management System (or DBMS) is essentially nothing more than a computerized data-keeping system.	IV
96	Data Models	--	Data modeling is the process of creating a simple diagram of a complex software system, using text and symbols to represent the way data will flow	IV
97	Object Relation Mapping	--	It is a programming technique for converting data between incompatible type systems using object-oriented programming languages.	IV
98	User Interface	UI	It is the process designers use to build interfaces in software or computerized devices, focusing on looks or style.	IV
99	Object Oriented User Interface	OOUI	In object-oriented user interface (OOUI) is a specific type of software interface where users act on specific objects for specific properties.	IV

100	Model View Controller	MVC	Model–view–controller (usually known as MVC) is a software design pattern commonly used for developing user interfaces that divide the related program logic into three interconnected elements.	IV
Unit-V : Case Tools				
101	Data Definition Language	DDL	It is a computer language used to create and modify the structure of database objects in a database.	V
102	Data Manipulation Language	DML	It is a computer programming language used for adding (inserting), deleting, and modifying (updating) data in a database.	V
103	Computer Aided Software Engineering	CASE	Computer-aided software engineering (CASE) is the domain of software tools used to design and implement applications. ...	V
104	Open Sources Software	--	It is the process by which open-source software, or similar software whose source code is publicly available, is developed by an open-source software project.	V
105	StarUML	--	StarUML is an open-source software modeling tool that supports the UML (Unified Modeling Language) framework for system and software modeling.	V
106	Rational Rose	--	Rational Rose is an object-oriented Unified Modeling Language (UML) software design tool intended for visual modeling and component construction of enterprise-level software applications.	V
107	Automatic Teller Machine	ATM	ATM is automatic teller machine which is basically used to withdraw money from an account	V
108	Withdraw	--	It is the act of taking something out, or stopping participation in an activity.	V
109	Deposit	--	A deposit is a financial term that means money held at a bank.	V
110	Stock Maintenance System	--	Stock maintenance system is basically for the customers who access the information about the stock and retrieves the information.	V
111	Quiz System	--	Quiz Management System Data flow diagram is often used as a preliminary step to create an overview of the Quiz without going into great detail, .	V
112	E-Mail Client System	--	An email client, email reader or, more formally, message user agent (MUA) or mail user agent is a computer program used to access and manage a user's email.	V
113	Cryptanalysis	--	It is the study of ciphertext, ciphers and cryptosystems with the aim of understanding how they work and finding and improving techniques for defeating or weakening them.	V
114	Health Care System	--	A health system, also sometimes referred to as a health care system, is the organization of people, institutions, and resources that deliver health care services to meet the health needs of target	V

			populations.	
115	Interaction Schema	--	Interactions contain messages that are exchanged between objects. ... The messages may also include creation and destruction of other objects.	V
116	Process Abstraction Controller	PAC	The controller is defined as the first object beyond the UI layer that receives and coordinates ("controls") a system operation.	V
117	Client	--	A client is somebody who buys goods or pays for services.	V
118	Server	--	A server is a computer or system that provides resources, data, services, or programs to other computers, known as clients, over a network	V
119	Concurrency	--	Concurrency allows more than one objects to receive events at the same time and more than one activity to be executed simultaneously.	V
120	Semantic Object Model	SOM	A named collection of attributes that sufficiently describes a distinct entity	V
121	Customer	--	A person who buys goods or services from a shop or business.	V
122	ArgoUML	--	ArgoUML is an open-source developer tool that helps in creating complex designs and diagrams.	V
123	Structural Diagram	--	It represents the static aspect of the system and the static parts of the diagrams are represented by classes, interfaces, objects, components and nodes	V
124	ObjectOriented Database Management System	OODBMS	In object-oriented database management system (OODBMS) applies concepts of object-oriented programming, and applies them to the management of persistent objects on behalf of multiple users, with capabilities for security, integrity, recovery and contention management.	V
125	Webserver	--	A web server is a computer that stores web server software and a website's component files.	V
PLACEMENT QUESTIONS				
126	Meta Models	--	It is model's model that serves for explanation and definition of relationships among the various components of the applied model itself.	
127	Containment	--	It is the idea that an outer class contains an instance of another class and allows access to the contained object through its own methods.	
128	Inception	--	This is the part of the project where the original idea is developed.	
129	Manipulators	--	Manipulators are helping functions that can modify the input/output stream	
130	Super class	--	A superclass or base class is a class that acts as a parent to some other class or classes.	
131	Sub class	--	A class that inherits from another class is called the subclass.	
132	Interface	--	Interface is a collection of methods of a class or component. It specifies the set of services that may be provided by the class or component.	

133	Access specifiers	--	Access modifiers (or access specifiers) are keywords in object-oriented languages that set the accessibility of classes, methods, and other members.	
134	Link	--	A link represents a connection through which an object collaborates with other objects.	
135	Relationships	--	Relationships in UML are used to represent a connection between structural, behavioral, or grouping things.	
136	Derived Attribute	--	Derived property is property which value (or values) is produced or computed from other information	
137	Multi-Valued Attribute	--	A multivalued attribute of an entity is an attribute that can have more than one value associated with the key of the entity.	
138	Role name	--	A role name identifies an end of an association and ideally describes the role played by objects in the association.	
139	Specialization	--	Specialization means creating new subclasses from an existing class.	
140	Data Dictionary	--	A data dictionary is a file or a set of files that includes a database's metadata.	
141	Object Oriented Programming	--	It is a programming paradigm based upon objects (having both data and methods) that aims to incorporate the advantages of modularity and reusability.	
142	Collaborators	--	It is another class that a class interacts with to fulfill its responsibilities.	
143	Elaboration	--	It is the initial series of iterations during which the team does serious investigation, implements (programs and tests) the core architecture, clarifies most requirements, and tackles the high-risk issues.	
144	Pattern	--	A pattern (or design pattern) is a written document that describes a general solution to a design problem that recurs repeatedly in many projects.	
145	Framework	--	A framework is a collection of coherent classes co-operating to solve a problem in a particular domain	
146	Factory Method	--	Factory Method is a creational design pattern that provides an interface for creating objects in a superclass, but allows subclasses to alter the type of objects that will be created.	
147	Bridge	--	Bridge is used when we need to decouple an abstraction from its implementation so that the two can vary independently.	

148	Adapter	--	It converts the interface of a class into another interface that a client wants"	
149	Observer	--	It is a software design pattern in which an object, named the subject, maintains a list of its dependents, called observers, and notifies them automatically of any state changes, usually by calling one of their methods.	
150	Creator	--	Creator refers to the object that instantiates a class.	

M.Menakapriya
Faculty Prepared   **Signature**

HoD