Roll No. : …………………………

Bharati Vidyapeeth's

Institute of Computer Applications and Management (BVICAM)

A-4, Paschim Vihar, New Delhi-63

SECOND SEMESTER [MCA] Internal Examination, May 2023

Paper Code: MCA–104	**Subject: Object Oriented Software Engineering**

Time: 2 Hours **Maximum Marks: 45**

Note: Attempt THREE questions in all. Question No. 1 is compulsory, and attempt one question from each unit.

1.	Answer all the following questions briefly: - 1.5 × 10 = 15		
	(a)	Enlist the different kinds of projects, a project manager is expected to manage? *The project manager is the most important element in the definition and execution of project management tasks. The success or failure of a given project is directly related to the skills and abilities of the project manager. A project manager may have both internal and external responsibilities.* *From, internal perspective, the project manager serves as the director and locus of control for the project team and all of the project activities. He establishes the team infrastructure so that the project can be accomplished.* *Few of the main responsibilities are:* *i. Identify the project tasks and build a work breakdown structure* *ii. Develop the project schedule* *iii. Recruit and train team members* *iv. Assign team members to tasks* *v. Coordinate activities of team members and sub teams* *vi. Assess project risks* *vii. Monitor and control project deliverables and milestones* *viii. Verify the quality of project deliverables* *From an external perspective, the project manager is the focal point or main contact for the project. He or she must represent the needs of the project and the team to the outside world. Some of the major external responsibilities include:* *i. Report project status and progress* *ii. Establish the working relationships with those providing the system requirements (i.e. the users)* *iii. Work with the client and other stakeholders* *iv. Identify resource needs and obtain resources*	CO2
	(b)	"UML is a process that produces models." Elaborate a comparison between the terms process and model? *A process is a set of partially ordered steps intended to reach a goal. UML is largely process-independent, i.e. you can use it with a number of software engineering processes. RUP is one such life cycle approach that is suited to the UML.* *A model helps us to visualize and understand a real life situation alongwith its behaviour.*	CO2
	(c)	What end result is expected at the end of the inception phase? *During the inception phase an overall project schedule is developed. Thus at the end of the inception phase a detailed project schedule for the next iteration is developed. During iteration a project schedule is used to track progress. Status reports and progress reports are calculated and the completion of the tasks and milestones can be compared with expected end dates. If the project goes off schedule the project manger must identify the causes and take corrective action to get the project back on schedule or at least stay current with the upcoming target dates.*	CO2

	(d)	Elaborate the significance of critical path in a PERT chart? *The critical path is the minimum time it will take to complete a project, based on the longest path from start to finish. It is the path where ECT=LCT*	CO2
	(e)	Enlist the different adornments to an association? *Most elements in the UML have a unique and direct graphical notation that provides a visual representation of the most important aspects of the element. For example, the notation for a class is intentionally designed to be easy to draw, because classes are the most common element found in modeling object-oriented systems. The class notation also exposes the most important aspects of a class, namely its name, attributes, and operations. A class's specification may include other details, such as whether it is abstract or the visibility of its attributes and operations. Many of these details can be rendered as graphical or textual adornments to the class's basic rectangular notation.*	CO1
	(f)	Why are UML models also called metadata?	CO1
	(g)	State the importance of specification in UML? *The UML is more than just a graphical language. Rather, behind every part of its graphical notation there is a specification that provides a textual statement of the syntax and semantics of that building block. For example, behind a class icon is a specification that provides the full set of attributes, operations (including their full signatures), and behaviors that the class embodies; visually, that class icon might only show a small part of this specification. Furthermore, there might be another view of that class that presents a completely different set of parts yet is still consistent with the class's underlying specification. You use the UML's graphical notation to visualize a system; you use the UML's specification to state the system's details. Given this split, it's possible to build up a model incrementally by drawing diagrams and then adding semantics to the model's specifications, or directly by creating a specification, perhaps by reverse engineering an existing system, and then creating diagrams that are projections into those specifications. The UML's specifications provide a semantic backplane that contains all the parts of all the models of a system, each part related to one another in a consistent fashion.*	CO1
	(h)	Justify whether UML is a closed language or open? *UML is an open language as it allows extensions*	CO1
	(i)	Explain the aggregation relationship in contrast to generalization?	CO1
	(j)	Elaborate the need of varied objects in the analysis model?	CO2

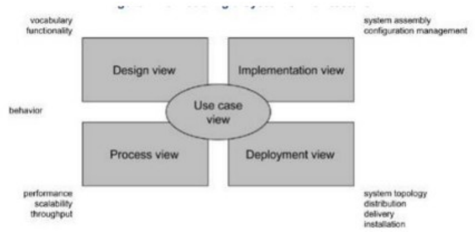
UNIT - I

2. (a) Compare traditional software development life cycle models with object oriented life cycle models? 5 CO1

FEATURE	*TRADITIONAL MODELS*	*OBJECT ORIENTED MODE*
Requirements Specification	*Generally done in beginning*	*Frequently changed*
Understanding Requirements	*Generally not well understood*	*Well understood*
Resource Control	*Yes mostly*	*No*
Cost Control	*Maybe*	*For Sure*
Simplicity	*Simple to intermediate*	*Complex*
Success Guarantee	*Low to medium*	*High*
% of Failures	*High*	*Low*

Methodology	*Functional and Process Driven*	*Object Driven*
Analysis Phase	*DFD, ER, Data Dictionary*	*Object Identification and description of attributes and operations*
Design Phase	*Structure chart, flowchart, pseudocode*	*Class diagram, object diagram, sequence diagram, collaboration diagram*

	(b)	Through suitable example elaborate the different building blocks of the UML? *Explain Things, relationships and diagrams*	5	CO1
	(c)	Explain the object oriented view of system architecture? *Jacobson Diagram*	5	CO1
3.	(a)	Justify the statement, "Architecture of a software intensive system can be best described by 5 interlocking views"?	5	CO1



The use case view *of a system encompasses the use cases that describe the behavior of the system as seen by its end users, analysts, and testers. This view doesn't really specify the organization of a software system. Rather, it exists to specify the forces that shape the system's architecture. With the UML, the static aspects of this view are captured in use case diagrams; the dynamic aspects of this view are captured in interaction diagrams, statechart diagrams, and activity diagrams.*

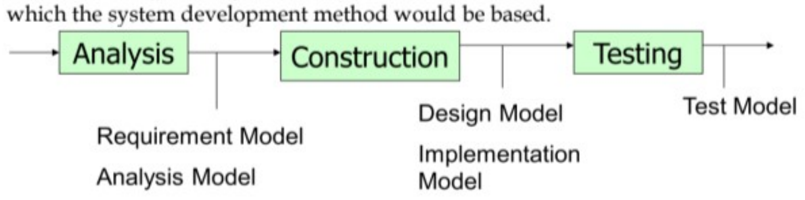
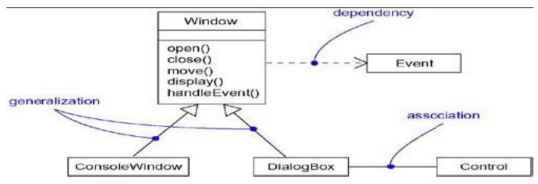
The design view *of a system encompasses the classes, interfaces, and collaborations that form the vocabulary of the problem and its solution. This view primarily supports the functional requirements of the system, meaning the services that the system should provide to its end users. With the UML, the static aspects of this view are captured in class diagrams and object diagrams; the dynamic aspects of this view are captured in interaction diagrams, statechart diagrams, and activity diagrams.*

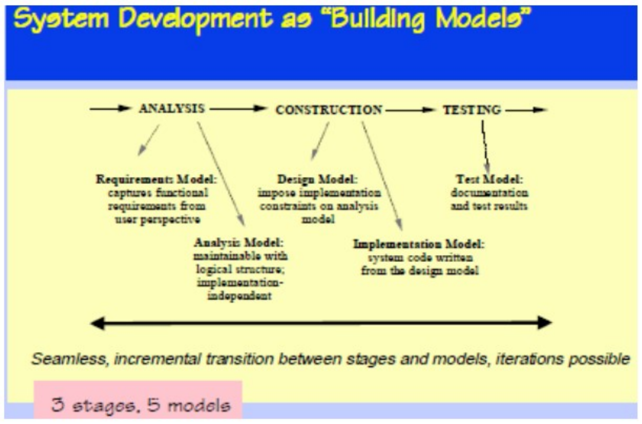
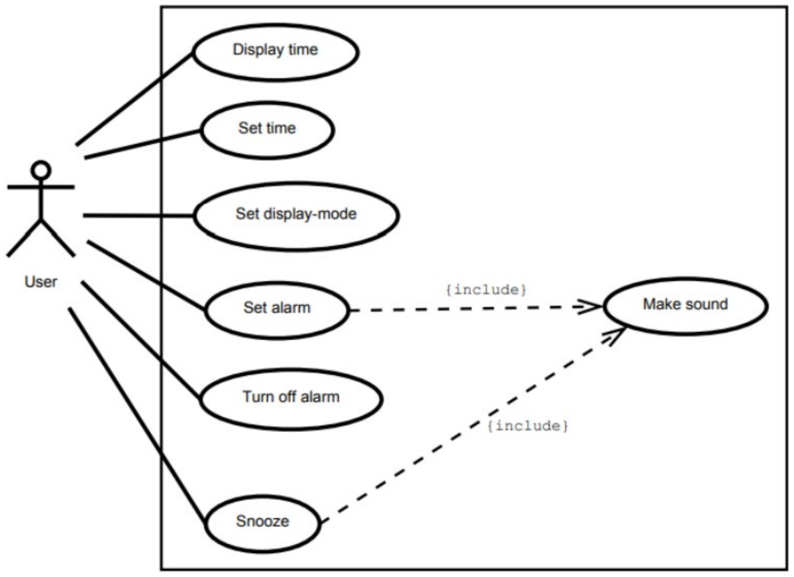
The process view *of a system encompasses the threads and processes that form the system's concurrency and synchronization mechanisms. This view primarily addresses the performance, scalability, and throughput of the system. With the UML, the static and dynamic aspects of this view are captured in the same kinds of diagrams as for the design view, but with a focus on the active classes that represent these threads and processes.*

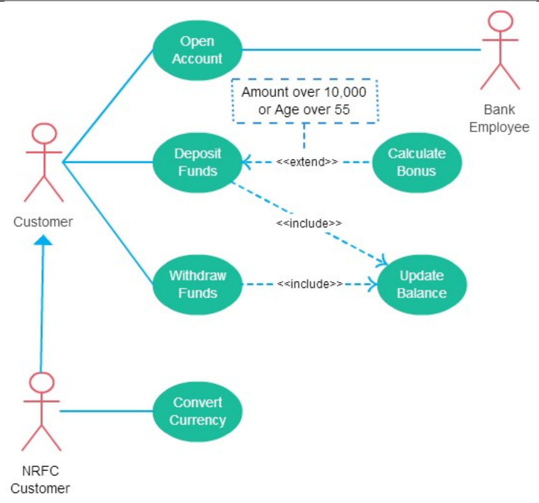
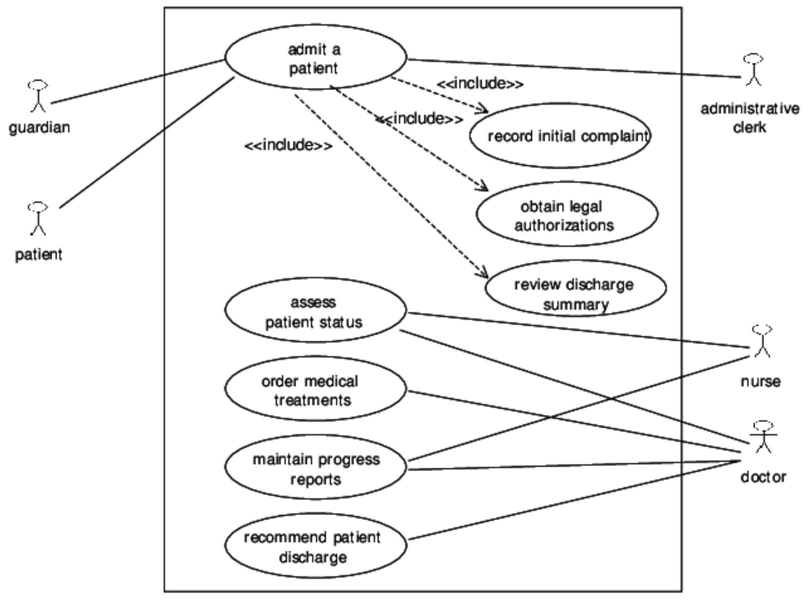
The implementation view *of a system encompasses the components and files that are used to assemble and release the physical system. This view primarily addresses the configuration management of the system's releases, made up of somewhat independent components and files that can be assembled in various ways to produce a running system. With the UML, the static aspects of this view are captured in component diagrams; the dynamic aspects of this view are captured in interaction diagrams, statechart diagrams, and activity diagrams.*

The deployment view *of a system encompasses the nodes that form the system's hardware topology on which the system executes. This view primarily addresses the distribution, delivery, and installation of the parts that make up the physical system. With the UML, the static aspects of this view are captured in deployment diagrams; the dynamic aspects of this view are captured in interaction diagrams, statechart diagrams, and activity diagrams.*

	(b)	Differentiate between a process and model?	5	CO1

		technique for each of these models. These modeling techniques (1 for each model)define the architecture on which the system development method would be based.  Analysis Construction Testing Requirement Model Analysis Model Design Model Implementation Model Test Model		
	(c)	Explain the different relationships possible in a UML diagram? *In the UML, the ways that things can connect to one another, either logically or physically, are modeled as relationships. In object-oriented modeling, there are three kinds of relationships that are most important: dependencies, generalizations, and associations. Dependencies are using relationships. For example, pipes depend on the water heater to heat the water they carry. Generalizations connect generalized classes to more-specialized ones in what is known as subclass/superclass or child/parent relationships. For example, a bay window is a kind of window with large, fixed panes; a patio window is a kind of window with panes that open side to side. Associations are structural relationships among instances. For example, rooms consist of walls and other things; walls themselves may have embedded doors and windows; pipes may pass through walls. These three kinds of relationships cover most of the important ways in which things collaborate with one another. Not surprisingly, they also map well to the ways that are provided by most object-oriented programming languages to connect objects. These three kinds of relationships cover most of the important ways in which things collaborate with one another. Not surprisingly, they also map well to the ways that are provided by most object-oriented programming languages to connect objects.*  Window open() close() move() display() handleEvent() dependency Event generalization association ConsoleWindow DialogBox Control Figure :relationships	5	CO1
UNIT - II				
4.	(a)	Justify the statement, "System development is based on model building"? *System development is a complex task. Many aspects should be considered to achieve a system i.e. reliable and performs its tasks properly. Thus we need to handle this complexity in an organized way. This is done by working with different models, each focusing on certain aspects of the system. By introducing complexity in a specific order in successive models, we are able to manage system complexity. There are 5 different models:* *i. Requirements Model: aims to capture the functional requirements. (mostly through the use case model).* *ii. Analysis Model: gives the system a robust and changeable object structure.(object diagram, class diagram)* *iii. Design Model: adopts and refines object structure to the current implementation environment.* *iv. Implementation Model:aims to implement the system*	5	CO2

		v. Test Model: aims to verify the system  System Development as "Building Models" ANALYSIS CONSTRUCTION TESTING Requirements Model: captures functional requirements from user perspective Analysis Model: maintainable with logical structure; implementation-independent Design Model: impose implementation constraints on analysis model Implementation Model: system code written from the design model Test Model: documentation and test results Seamless, incremental transition between stages and models, iterations possible 3 stages, 5 models		
	(b)	Suppose we want to develop software for an alarm clock. The clock shows the time of day. Using buttons, the user can set the hours and minutes fields individually, and choose between 12 and 24-hour display. It is possible to set one or two alarms. When an alarm fires, it will sound some noise. The user can turn it off, or choose to 'snooze'. If the user does not respond at all, the alarm will turn off itself after 2 minutes. 'Snoozing' means to turn off the sound, but the alarm will fire again after some minutes of delay. This 'snoozing time' is pre-adjustable. Identify the top-level functional requirement for the clock, and model it with a use case diagram. Describe in brief each use case?  Display time Set time Set display-mode Set alarm Turn off alarm Snooze Make sound {include} {include} User	1 0	CO2
5.	(a)	Which association can be viewed as an interrupt in the original use case? *The include association can be viewed as an interrupt in the original use case. Include relationship show that the behavior of the included use case is part of the including (base) use case. The main reason for this is to reuse common actions across multiple use cases. In some situations, this is done to simplify complex behaviors. Few things to consider when using the <<include>> relationship.* • *The base use case is incomplete without the included use case.* • *The included use case is mandatory and not optional.*	5	CO2

		 Open Account Bank Employee Amount over 10,000 or Age over 55 Customer Deposit Funds <<extend>> Calculate Bonus <<include>> Withdraw Funds <<include>> Update Balance Convert Currency NRFC Customer		
	(b)	A patient record and scheduling system in a doctor's office is used by the receptionists, nurses, and doctors. The receptionists use the system to enter new patient information when first-time patients visit the doctor. They also schedule all appointments. The nurses use the system to keep track of the results of each visit including diagnosis and medications. For each visit, free form text fields are used captures information on diagnosis and treatment. Multiple medications may be prescribed during each visit. The nurses can also access the information to print out a history of patient visits. The doctors primarily use the system to view patient history. The doctors may enter some patient treatment information and prescriptions occasionally, but most frequently they let the nurses enter this information. -- Each patient is assigned to a family. The head of family is responsible for the person with the primary medical coverage. Information about doctors is maintained since a family has a primary care physician, but different doctors may be the ones seeing the patient during the visit. Identify the top-level functional requirement for the clock, and model it with a use case diagram. Describe in brief each use case?  admit a patient guardian patient <<include>> <<include>> <<include>> record initial complaint obtain legal authorizations review discharge summary administrative clerk assess patient status order medical treatments maintain progress reports recommend patient discharge nurse doctor	1 0	CO2
