OBJECT ORIENTED SYSTEM AND DESIGN

QUESTION BANK

Q1. Differentiate synchronous and asynchronous messages.

Ans: Synchronous messages, which are associated with an operation, have a send and a receive message. A message is sent from the source lifeline to the target lifeline. The source lifeline is blocked from other operations until it receives a response from the target lifeline.

Asynchronous messages, which are associated with operations, typically have only a send message, but can also have a reply message. In contrast to a synchronous message, the source lifeline is not blocked from receiving or sending other messages. You can also move the send and receive points individually to delay the time between the send and receive events.

Q2. What do you mean by an object? Discuss the various characteristics of object with examples?

Ans:

Object is an instance of class. Class is an entity in which data and functions are organized. Each object has a class which defines its data and behavior. When program is executed the objects interact by sending messages to one another.

Object characterstics are:

-Encapsulation – Encapsulation is capturing data and keeping it safely and securely from outside interfaces.

-Inheritance- This is the process by which a class can be derived from a base class with all features of base class and some of its own. This increases code reusability.

-Polymorphism- This is the ability to exist in various forms. For example an operator can be overloaded so as to add two integer numbers and two floats.

-Abstraction- The ability to represent data at a very conceptual level without any details.

- Emphasis on data rather than procedure
- Programs are divided into entities known as objects
- Data Structures are designed such that they characterize objects
- Functions that operate on data of an object are tied together in data structures
- Data is hidden and cannot be accessed by external functions
- Objects communicate with each other through functions
- New data and functions can be easily added whenever necessary
- Follows bottom up design in program design

Q3. Discuss JSD and its approach comparing with object modeling techniques.

Ans: JSD, Jackson Structure Design is a methodology to specify and design systems in which time factor is significant and system may be described using sequence of events. Developed by Michael A. Jackson , this design method considers the fact that the design of the system is an extension of the programme design. The purpose of this design method is to create a maintainable software. The method addresses all stages of the software development life cycle. It has three phases:
Modeling phase - A JSD model starts with real world consideration. This phase is a part of analysis process. The aspects of the real world relevant to the system being developed are modeled in this phase.
Specification phase - This phase focuses on the specification. In this phase JSD determines what is to be done? The previous phase i.e. the modeling phase provides the basic for the system specifications to achieve the required functionality.
implementation phase - In this phase , JSD determines how to achieve required functionality. Operational specifications are executed so that it expresses desired system behavior in terms of some abstract machine.

Steps for JSD software development
Originally presented by Jackson in 1983 the method consisted of six steps which are following.

- Entity /action step
- Entity structure step
- Initial model step
- function step
- system timing step
- system implementing step

Q4. What are the three models in OMT? How object oriented analysis and design attached with OMT. Explain with example?

Ans The **object-modeling technique** (**OMT**) is an object modeling approach for software modeling and designing. It was developed around 1991 by Rumbaugh as a method to develop object-oriented systems and to support object-oriented programming. OMT describes object model or static structure of the system.

OMT was developed as an approach to software development. The purposes of modeling according to Rumbaugh are:

- testing physical entities before building them (simulation),
- communication with customers,
- visualization (alternative presentation of information), and
- reduction of complexity.

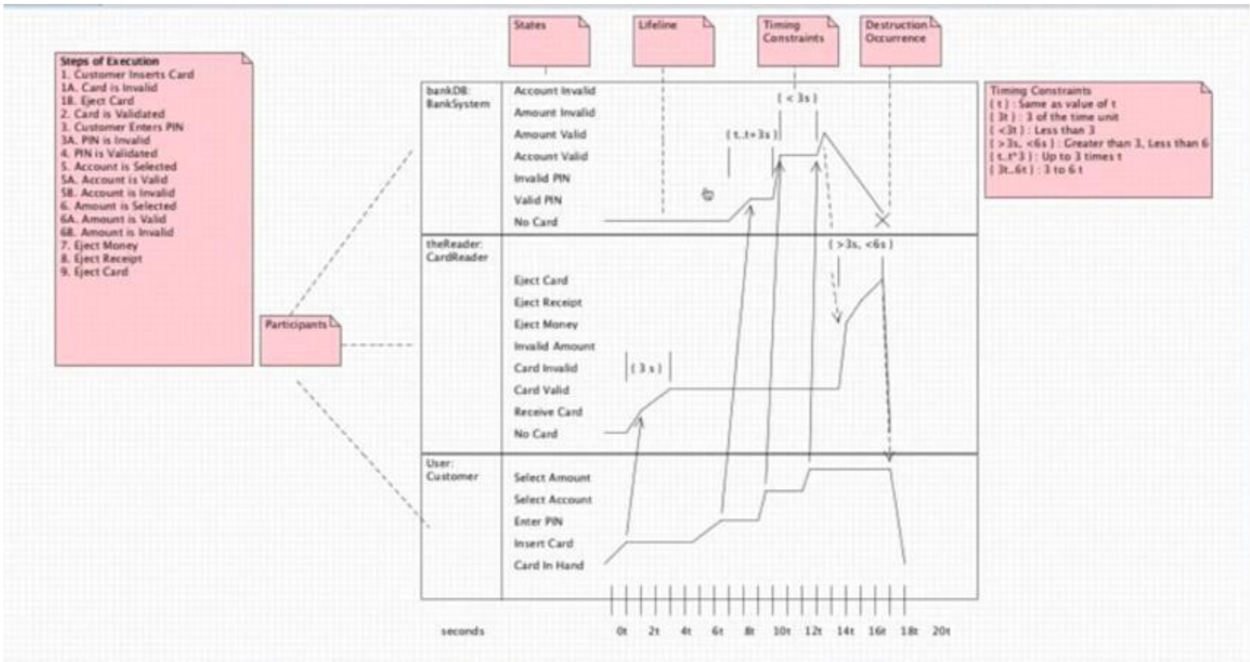
OMT has proposed three main types of models:

- ***Object*** *model*: The object model represents the static and most stable phenomena in the modeled domain. Main concepts are classes and associations with attributes and operations. Aggregation and generalization (with multiple inheritance) are predefined relationships.
- ***Dynamic*** *model*: The dynamic model represents a state/transition view on the model. Main concepts are states, transitions between states, and events to trigger transitions. Actions can be modeled as occurring within states. Generalization and aggregation (concurrency) are predefined relationships.
- ***Functional*** *model*: The functional model handles the process perspective of the model, corresponding roughly to data flow diagrams. Main concepts are process, data store, data flow, and actors.

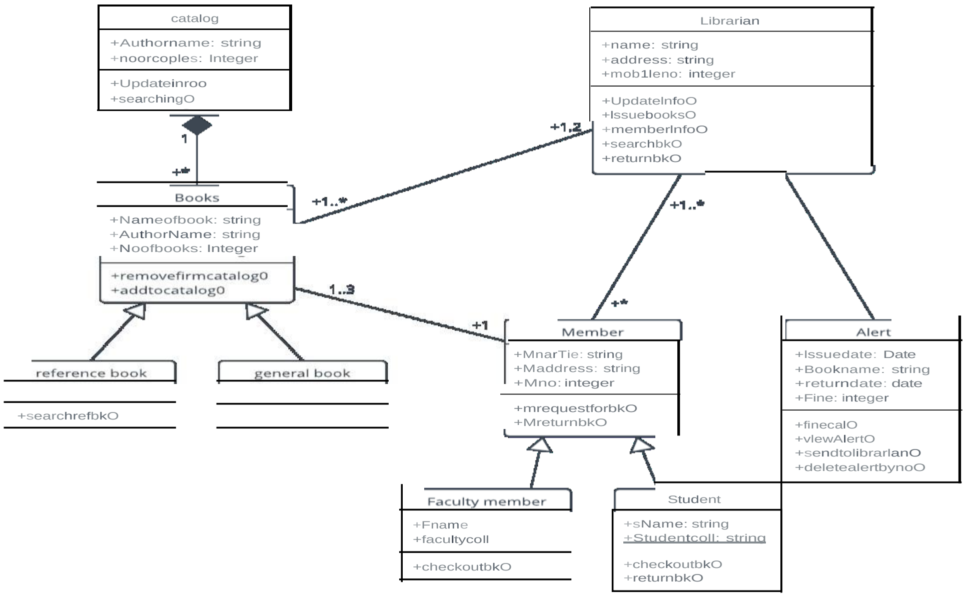
The systems designed using OMT are closer to the real world as the real world functioning of the system is directly mapped into the system designed using OMT. Because of this, it becomes easier to produce and understand designs.

• The objects in the system are immune to requirement changes because of data hiding and encapsulation features of object orientation.
• OMT designs encourage more reusability. The classes once defined can easily be used by other applications. This is achieved by defining classes and putting them into a library of classes where all the classes are maintained for future use. Whenever a new class is needed the programmer first looks into the library of classes and if it is available, it can be used as it is or with some modification. This reduces the development cost & time and increases quality.
• As the programmer has to spend less time and effort so he can utilize saved time in concentrating on other aspects of the system.

Q5. Draw UML Timing Diagram for ATM machine.



Q22. Draw UML Class Diagram for Library Management System.


catalog
+Authorname: string
+noorcoples: Integer
+Updateinroo
+searchingO
1
+*
Books
+Nameofbook: string
+AuthorName: string
+Noofbooks: Integer
+removefirmcatalog0
+addtocatalog0
+1..*
+1.2
1..3
+1
reference book
+searchrefbkO
general book
Librarian
+name: string
+address: string
+mob1leno: integer
+UpdateInfoO
+IssuebooksO
+memberInfoO
+searchbkO
+returnbkO
+1..*
+*
Member
+MnarTie: string
+Maddress: string
+Mno: integer
+mrequestforbkO
+MreturnbkO
Alert
+Issuedate: Date
+Bookname: string
+returndate: date
+Fine: integer
+finecalO
+vlewAlertO
+sendtolibrarlanO
+deletealertbynoO
Faculty member
+Fname
+facultycoll
+checkoutbkO
Student
+sName: string
+Studentcoll: string
+checkoutbkO
+returnbkO