CELEBRATING 25 YEARS

CMRIT

CMR INSTITUTE OF TECHNOLOGY, BENGALURU.

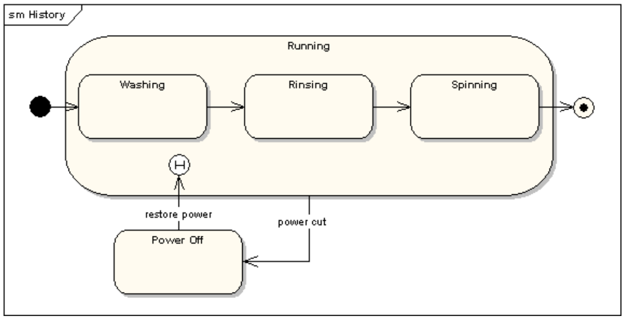
ACCREDITED WITH A+ GRADE BY NAAC

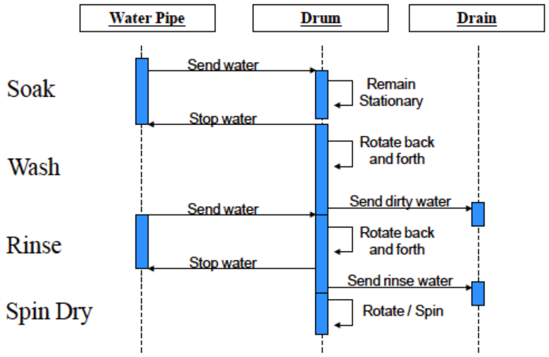
INTERNAL ASSESSMENT TEST 2 – Oct -Nov 2020

Scheme and Solution

Sub:	**SOFTWARE ENGINEERING**					Sub Code:	**18CS35**	Branch:	**ISE**
Date:	04-11-2020	Duration:	90 min's	Max Marks:	50	Sem / Sec:	III SEM C sec.		OBE

Answer any 5 Questions (5 X 10 = 50)

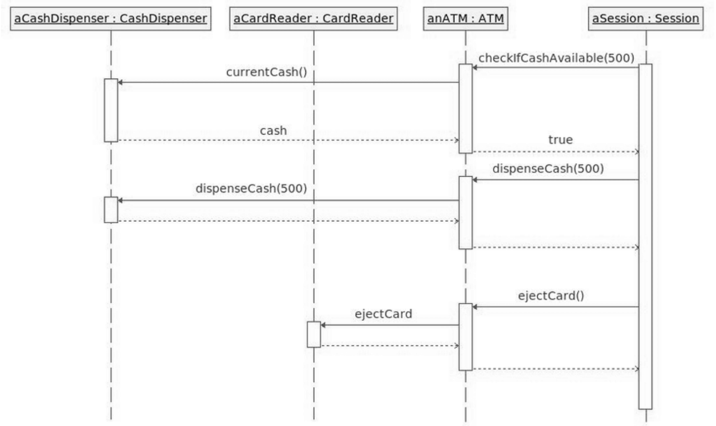
1.	a) With suitable diagrams, differentiate Interaction Model and State Model by taking example of Washing machine operation. Ans: **A State Model** describe all of the possible states that a particular object (or even the entire system) can get into. State Machines give us the means to control decisions and each state is like a 'mode of operation' for the object which behaves differently depending on its state.  sm History Running Washing Rinsing Spinning H restore power power cut Power Off Interaction Model: Interaction modeling describes how the different elements in an object model interact with each other. The term interacted is a set of exchanged messages between the objects.	**[3+3]**	**CO1**	**L3**



1. **State Model:**
State model describes those aspects of objects concerned with time and the sequencing of operations – events that mark changes, states that define the context for events, and the organization of events and states.Actions and events in a state diagram become operations on objects in the class model. State diagram describes the state model.

2. **Interaction Model:**
Interaction model is used to show the various interactions between objects, how the objects collaborate to achieve the behavior of the system as a whole.
The following diagrams are used to show the interaction model:
   - Use Case Diagram
   - Sequence Diagram
   - Activity Diagram

b) Design a sequence diagram for cash withdraw operation in ATM. [04] CO1 L3



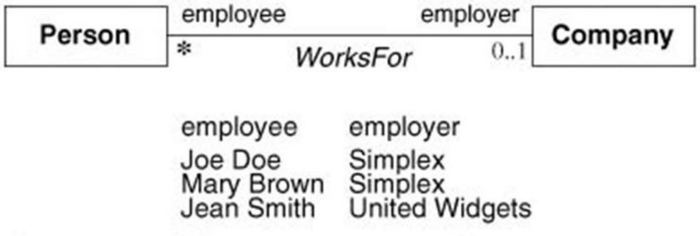
2. With the help of a sample class model of relation between Bank, Account number and bank customer, explain the following: [4 x 2.5=10] CO1 L2

1) Association and Association end name 2) Qualified association 3) Association classes 4) Multiplicity

Ans:

1. Association and Association end names:

Association end is a connection between the line depicting an association and the icon depicting the connected classifier. Name of the association end may be placed near the end of the line. The association end name is commonly referred to as **role** name (but it is not defined as such in the UML 2.4 standard). The role name is optional and suppressible.



employee	employer
Joe Doe	Simplex
Mary Brown	Simplex
Jean Smith	United Widgets

2. Qualified Assocoation: A qualified association is an association in which an attribute called the qualifier disambiguates the objects for a "many" association end.

- It is possible to define qualifiers for one-to many and many-to-many associations.
- A qualifier selects among the target objects, reducing the effective multiplicity, from "many" to "one.“

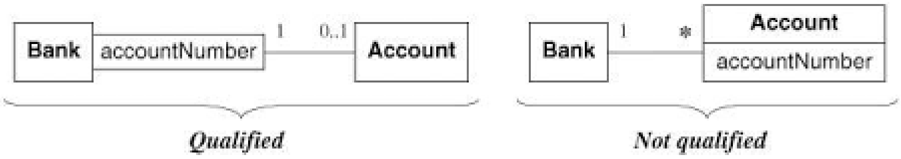
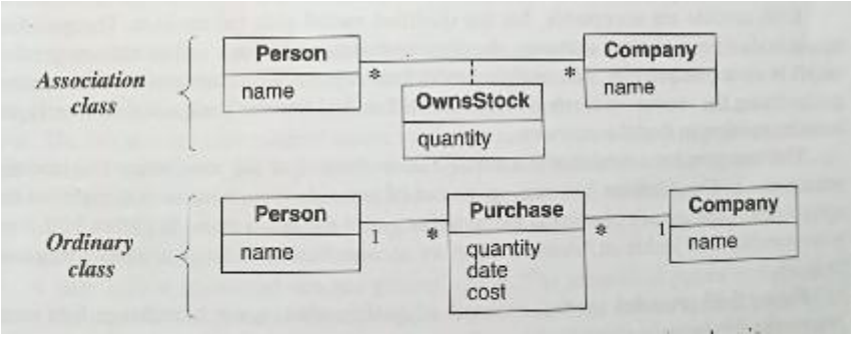
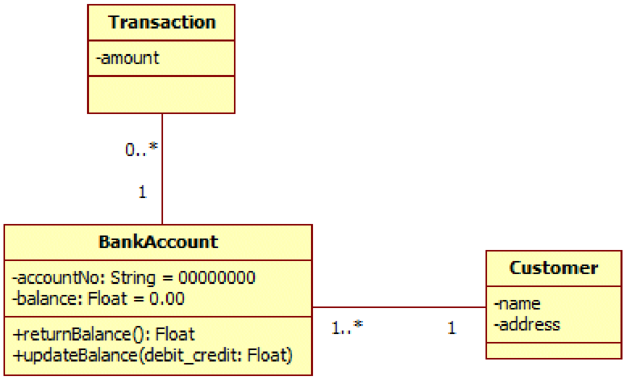
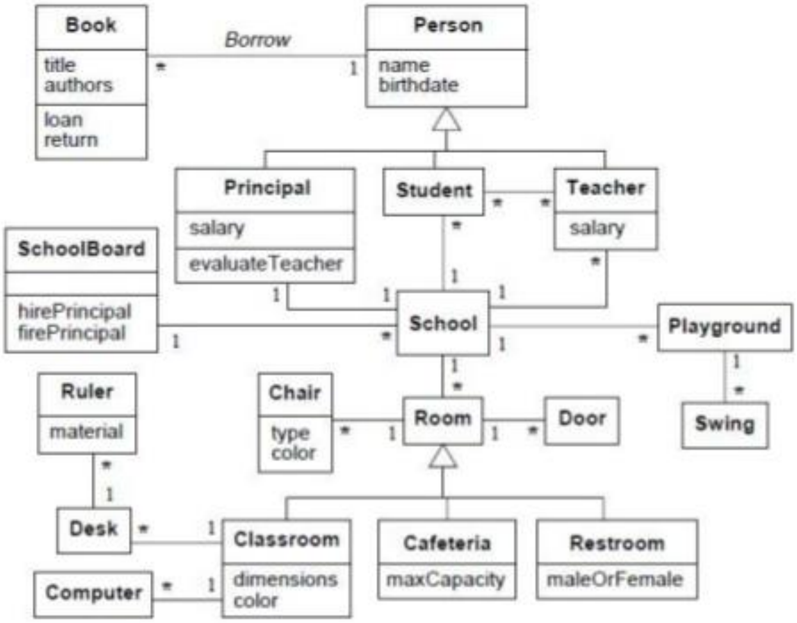


Figure 3.22 Qualified association. Qualification increases the precision of a model.

3. Association Class:



4. Multiplicity:

	 Transaction -amount 0..* 1 BankAccount -accountNo: String = 00000000 -balance: Float = 0.00 +returnBalance(): Float +updateBalance(debit_credit: Float) 1..* 1 Customer -name -address			
3.	Design a class diagram for the following group of classes: School, playground, principal, book, student, teacher, cafeteria, class room and computer. Use association names where ever needed and show multiplicity. Ans:  Book title authors loan return Borrow * 1 Person name birthdate Principal salary evaluateTeacher Student Teacher salary SchoolBoard hirePrincipal firePrincipal School Playground Swing Ruler material Desk Computer Chair type color Room Door Classroom dimensions color Cafeteria maxCapacity Restroom maleOrFemale	[10]	CO1	L3
4.	Explain the following Terms with respect to Object oriented programming: 1) Abstraction 2) Classification 3) Inheritance 4)Polymorphism Ans: 1. Abstraction: Abstraction One of the most fundamental concept of OOPs is Abstraction. Abstraction is a process where you show only “relevant” data and “hide” unnecessary details of an	[4 x 2.5=10]	CO1	L2

object from the user. For example, when you login to your Amazon account online, you enter your user id and password and press login, what happens when you press login, how the input data sent to amazon server, how it gets verified is all abstracted away from the you.

Another example of abstraction: A car in itself is a well-defined object, which is composed of several other smaller objects like a gearing system, steering mechanism, engine, which are again have their own subsystems. But for humans car is a one single object, which can be managed by the help of its subsystems, even if their inner details are unknown.

2. Inheritance:

This post provides the theoretical explanation of inheritance with real-life examples. For detailed explanation on this topic with java programs refer inheritance with examples and types of inheritance in java.

- Inheritance is the mechanism by which an object acquires the some/all properties of another object.
- It supports the concept of hierarchical classification.

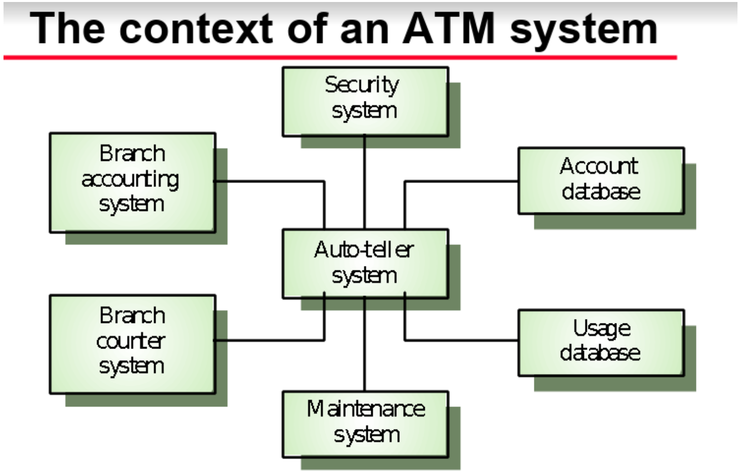
For example: Car is a four wheeler vehicle so assume that we have a class FourWheeler and a sub class of it named Car. Here Car acquires the properties of a class FourWheeler. Other classifications could be a jeep, tempo, van etc. FourWheeler defines a class of vehicles that have four wheels, and specific range of engine power, load carrying capacity etc. Car (termed as a sub-class) acquires these properties from FourWheeler, and has some specific properties, which are different from other classifications of FourWheeler, such as luxury, comfort, shape, size, usage etc.

A car can have further classification such as an open car, small car, big car etc, which will acquire the properties from both Four Wheeler and Car, but will still have some specific properties. This way the level of hierarchy can be extended to any level.

3. Polymorphism:

This post provides the theoretical explanation of polymorphism with real-life examples. For detailed explanation on this topic with java programs refer polymorphism in java and runtime & compile time polymorphism.

- Polymorphism means to process objects differently based on their data type.
- In other words it means, one method with multiple implementation, for a certain class of action. And which implementation to be used is decided at runtime depending upon the situation (i.e., data type of the object)
- This can be implemented by designing a generic interface, which provides generic methods for a certain class of action and there can be multiple classes, which provides the implementation of these generic methods.

	Lets us look at same example of a car. A car have a gear transmission system. It has four front gears and one backward gear. When the engine is accelerated then depending upon which gear is engaged different amount power and movement is delivered to the car. The action is same applying gear but based on the type of gear the action behaves differently or you can say that it shows many forms (polymorphism means many forms) Polymorphism could be static and dynamic both. Method Overloading is static polymorphism while, Method overriding is dynamic polymorphism. • Overloading in simple words means more than one method having the same method name that behaves differently based on the arguments passed while calling the method. This called static because, which method to be invoked is decided at the time of compilation • Overriding means a derived class is implementing a method of its super class. The call to overriden method is resolved at runtime, thus called runtime polymorphism			
5.	a)With neat diagram explain context model, with an example. Ans: 1. Context models are used to illustrate the boundaries of a system. 2. Social and organizational concerns may affect the decision on where to position system boundaries 3. Decision on what functionalities need to be included.  The context of an ATM system Security system Branch accounting system Account database Auto-teller system Branch counter system Usage database Maintenance system	[06]	CO1	L2
	b) Explain the four phases of Rational Unified Process Model. Ans: RUP is a phased model Identifies four discrete phases in software process.	[04]	CO1	L2

<table>
<tr>
<td></td>
<td>

1. Inception- business case for the system
2. Eloboration - understand problem domain, develop project plan
3. Construction - design, programming and testing
4. Transition - moving system from development community to the user community.

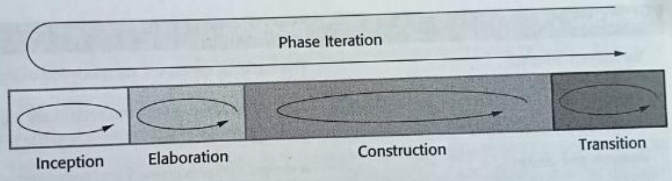


Figure 2.12 Phases in the Rational Unified Process

interact with the system and define these interactions. You then use this infor-
that the system makes to the business. If this

</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>

a)What is the difference between Data driven Model and Event driven Model? Write an Event Driven model for Microwave Oven operation.

Ans:

Data driven programming is a programming model where the data itself controls the flow of the program (not the program logic) where in case of Event driven programming , it is the event not the data itself controls the flow of the program.

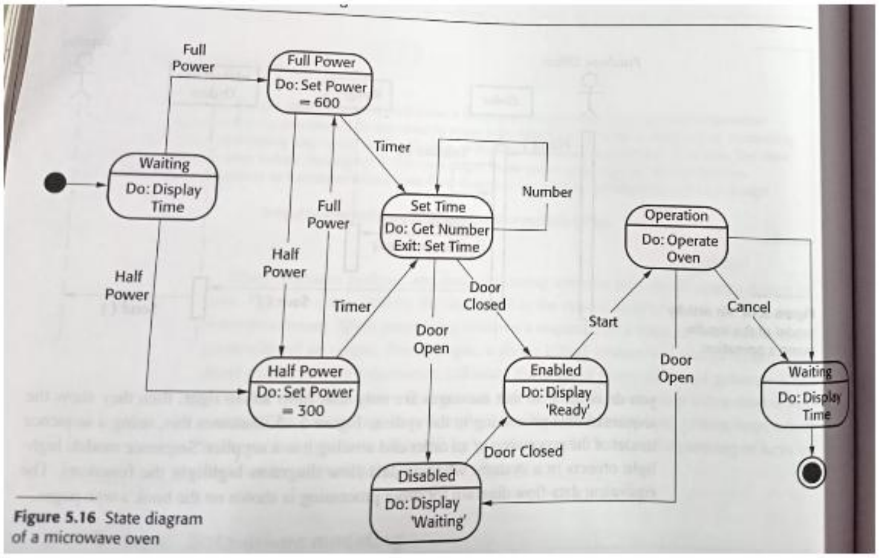
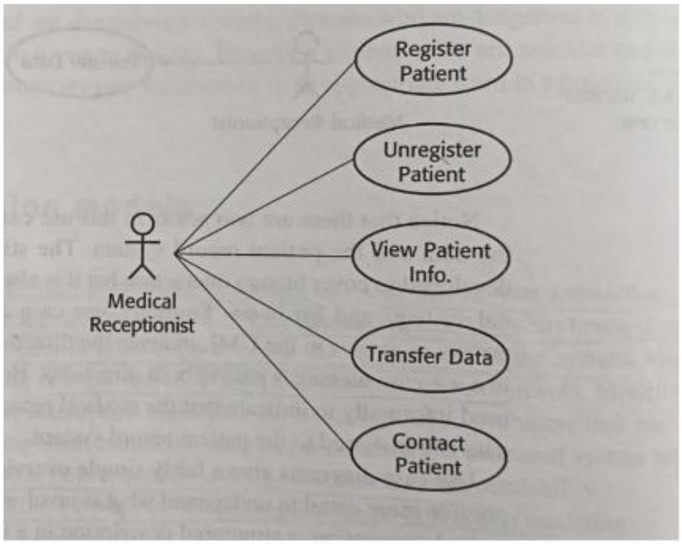
Data-driven modeling

- Many business systems are data-processing systems that are primarily driven by data.
- Controlled by the data input to the system, with relatively little external event processing.
- Data-driven models show the sequence of actions involved in processing input data and generating an associated output.
- They are particularly useful during the analysis of requirements as they can be used to show end-to-end processing in a system.

Event-driven modeling

- Real-time systems are often event-driven, with minimal data processing.
- Event-driven modeling shows how a system responds to external and internal events.
- Based on the assumption that a system has a finite number of states and that events (stimuli) may cause a transition from one state to another.

</td>
<td><b>[08]</b></td>
<td><b>CO1</b></td>
<td><b>L3</b></td>
</tr>
</table>

	 Full Power Full Power Do: Set Power = 600 Timer Waiting Do: Display Time Full Power Set Time Do: Get Number Exit: Set Time Number Operation Do: Operate Oven Half Power Half Power Timer Door Closed Cancel Start Door Open Door Open Half Power Do: Set Power = 300 Enabled Do: Display 'Ready' Waiting Do: Display Time Door Closed Disabled Do: Display 'Waiting' **Figure 5.16** State diagram of a microwave oven			
	b)Write the Usecases involving the role "Medical receptionist" with Use case diagram. Ans;  Register Patient Unregister Patient View Patient Info. Transfer Data Contact Patient Medical Receptionist	**[02]**	**CO1**	**L3**