SLIIT
Discover Your Future

SOFTWARE PROCESS MODELING

Activity Diagrams

SLIIT
Discover Your Future

Session Outcomes

- Introduction.
- Elements of Activity diagram.
- Partitioning an activity diagram.
- Applying activity diagrams in real world applications.

SLIIT
Discover Your Future

What is an activity diagram?

- An Activity Diagram models
  - Activities of a system.
  - Dependencies between activities.
  - Represents Workflows.

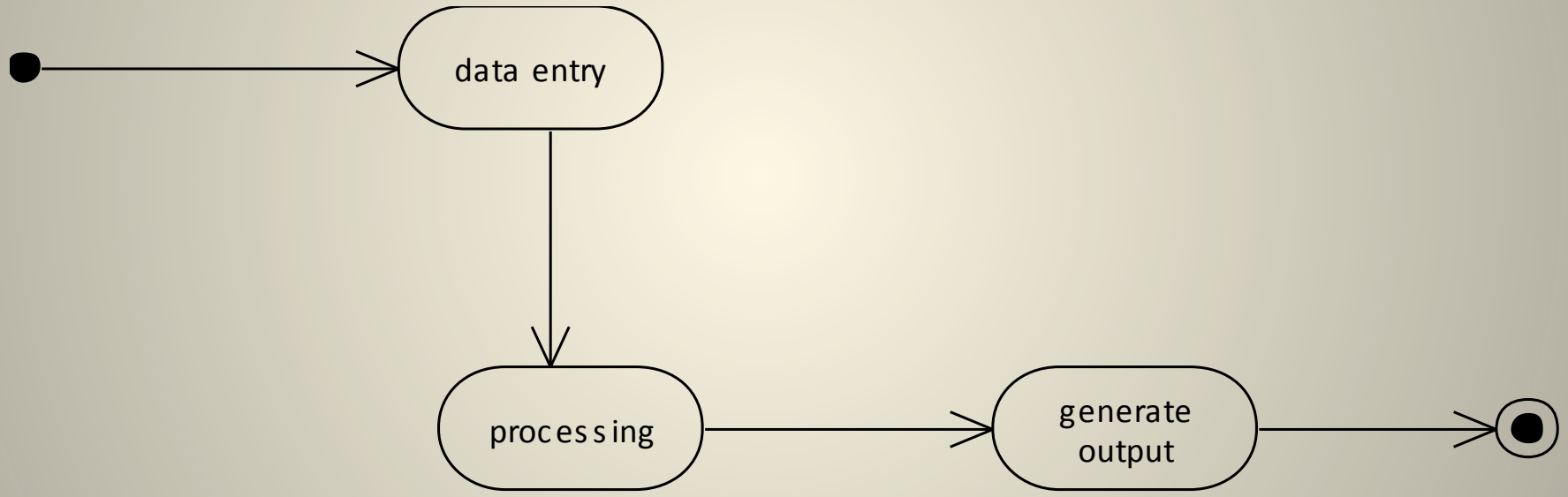
SLIIT
Discover Your Future

When to Use Activity Diagrams ?

- Modeling business **processes.**
- Analyzing **system functionality to** identify the use cases.
- Analyzing **individual use cases in** detail.
- Clarifying **concurrency issues**.

SLIIT
Discover Your Future

A simple Activity Diagram



SLIIT
Discover Your Future

Elements of an Activity Diagram

- **Action**
  - Simple Action
  - Call Action
- Transition
- Controls
  - Initial/Start
  - Final/End
  - Nodes
    - Decision/Branch
    - Merge
    - Fork
    - Join

SLIIT
Discover Your Future

Initial & Final Nodes

- Represent the beginning and end of a diagram respectively.
  - Initial/Start Node – Filled Circle
  - Final node is within circle

SLIIT
Discover Your Future

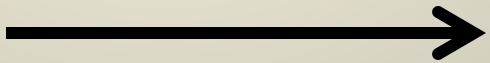
Action Node

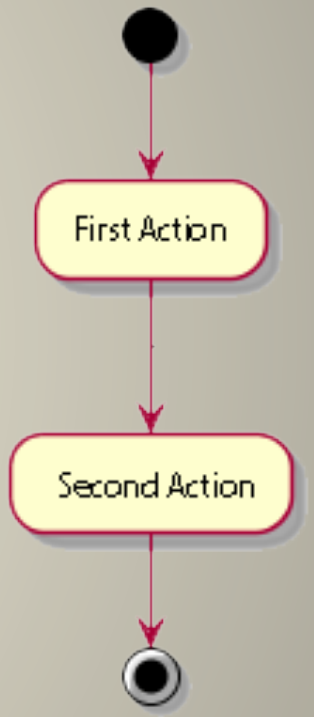
- An action is some task which needs to be done.
- Each action can be followed by another action .
- Represented with a **rounded rectangle.**
- Text in the action box should represent an activity (**verb phrase** in present tense).
- An activity is a sequence of actions.

SLIIT
Discover Your Future

Transition

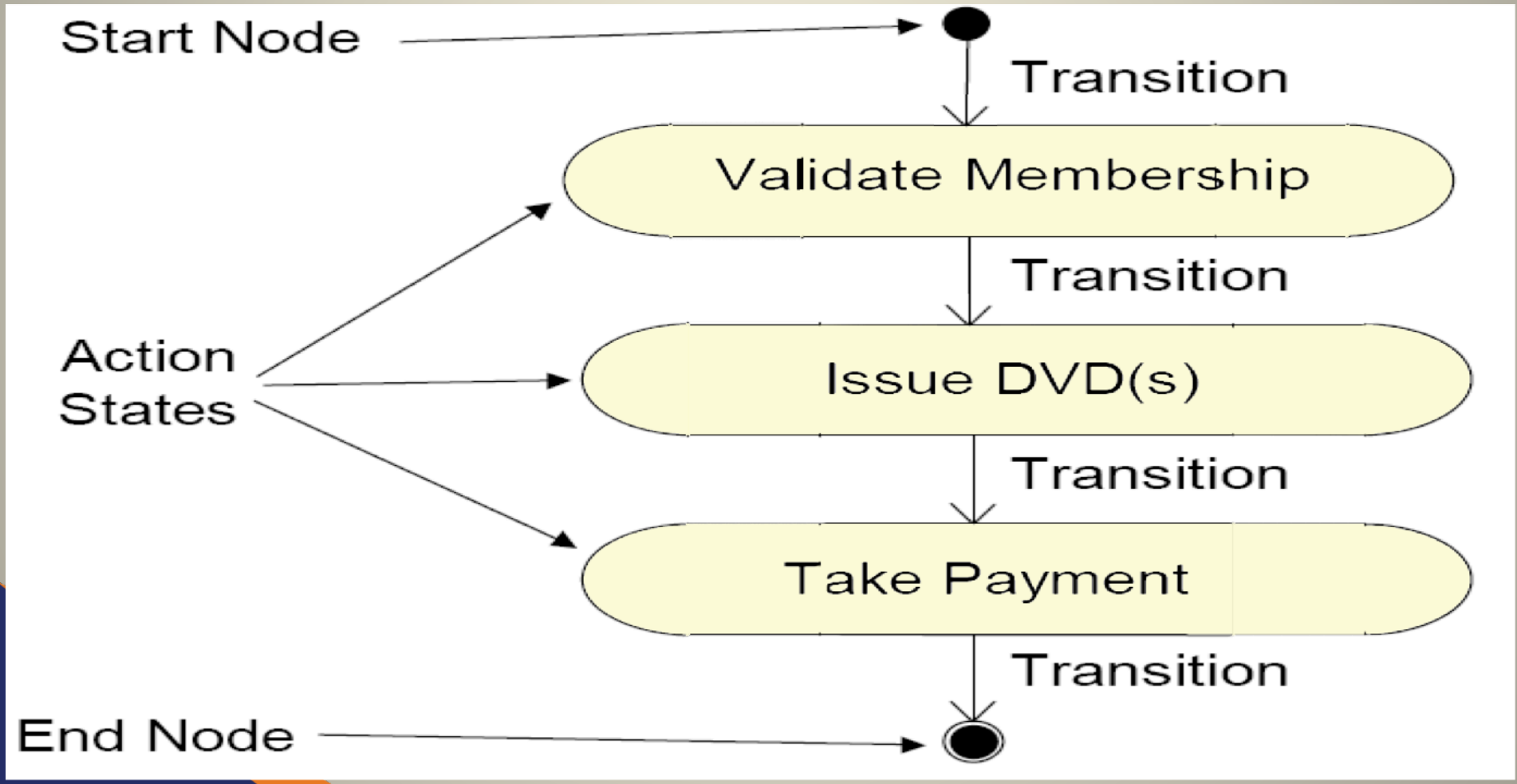
- Activity/Action nodes are connected by Transitions.
- Also known as a control flow/directed flow /edge
- When the execution of the node completes, execution proceeds to the node found on the output flow.





SLIIT
Discover Your Future

Activity Diagrams: Example



SLIIT
Discover Your Future

Activity - 1

Draw an activity diagram for becoming a member of the Library.

- To become a library member a student must keep a refundable security deposit of Rs.3000/=.
- Few registration forms should be filled in order to handle membership.
- Upon registration a member will be given a membership ID and a password.

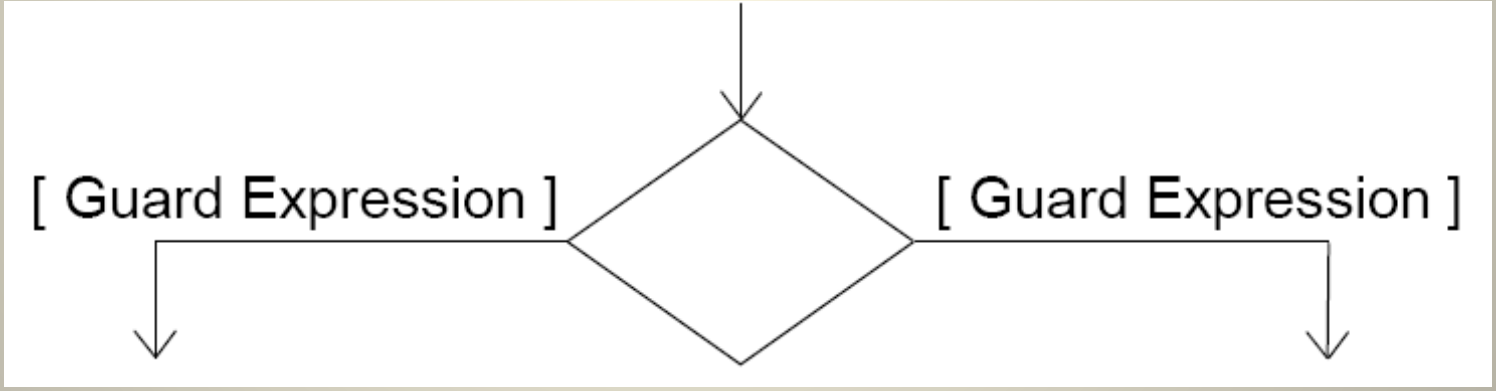
Decision Node

- A Decision represents a conditional flow of control:
  - the alternatives are mutually exclusive.
- Similar to IF/ELSE statements in programming languages like C/C++/Java.

SLIIT
Discover Your Future

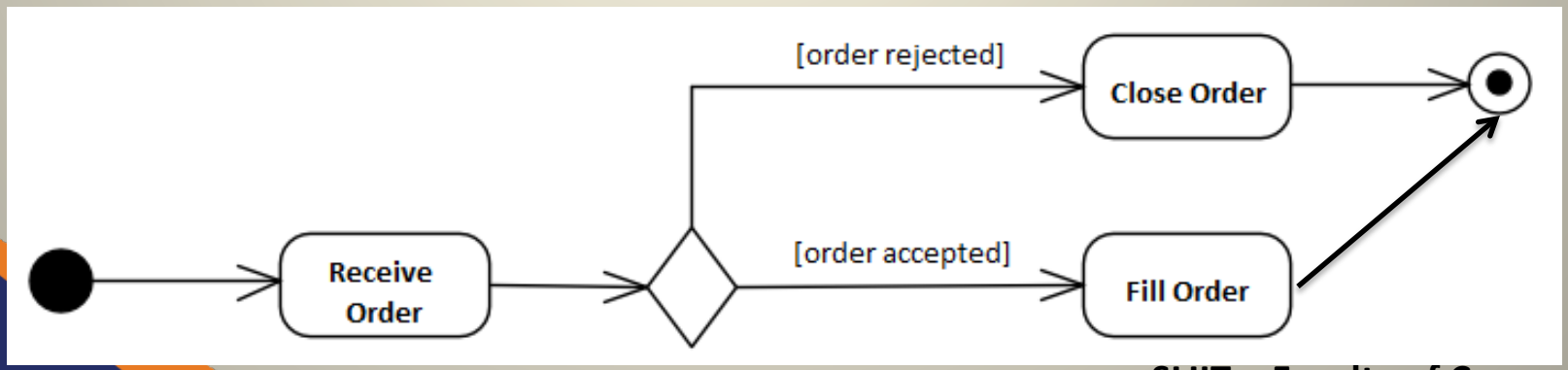
Decision Node

- A Decision/Branch is represented by:
  - a diamond at the branch point.
  - a guard expression for each possibility.



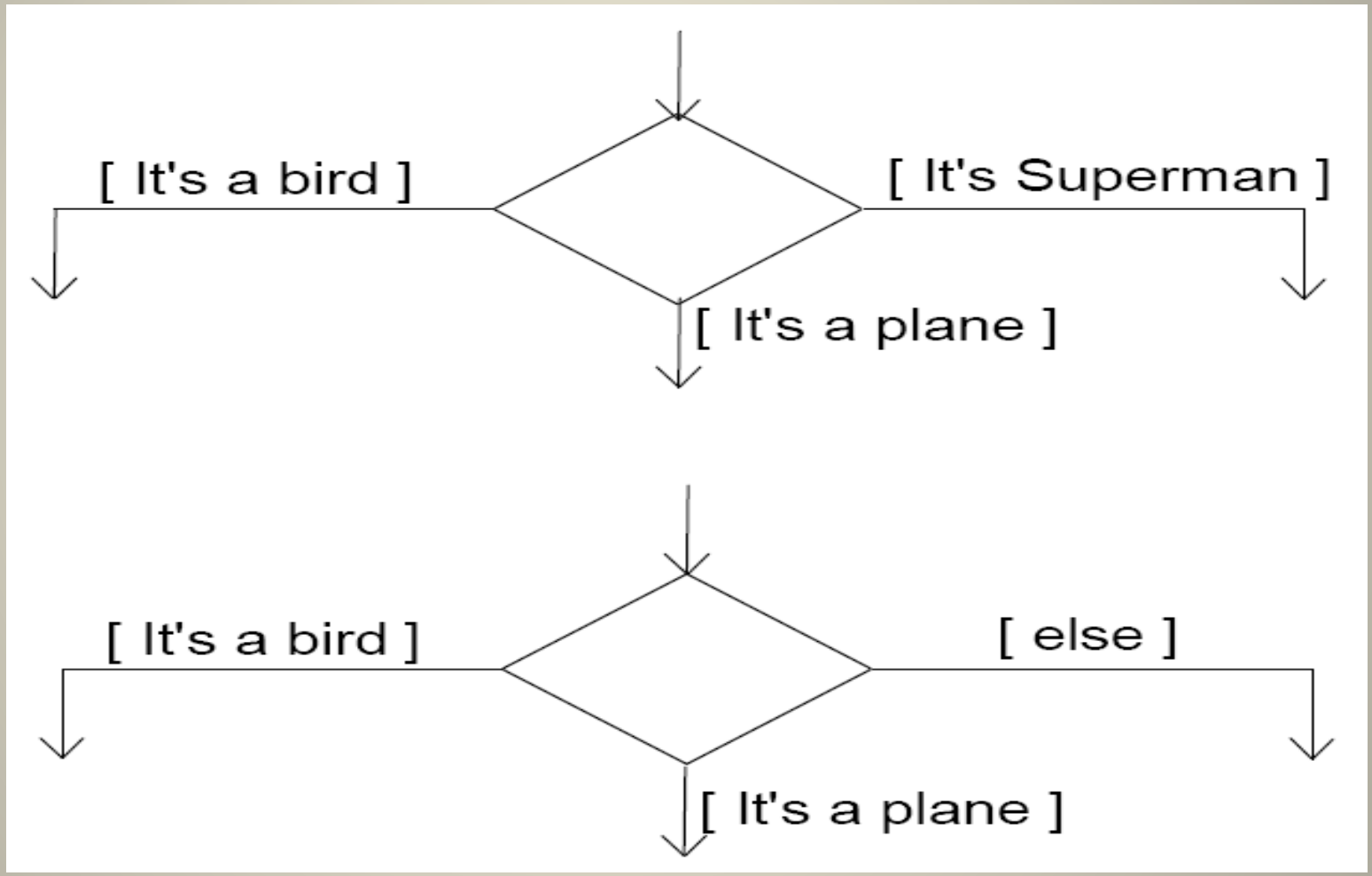
Decision Node

- Decision/Branch points.
  - Each branch must have a **guard condition**.
  - The **flow of control** flows down the **single** path where the branch condition is true.
  - There is **no limit** on the number of **branches** each branch point may have.



SLIIT
Discover Your Future

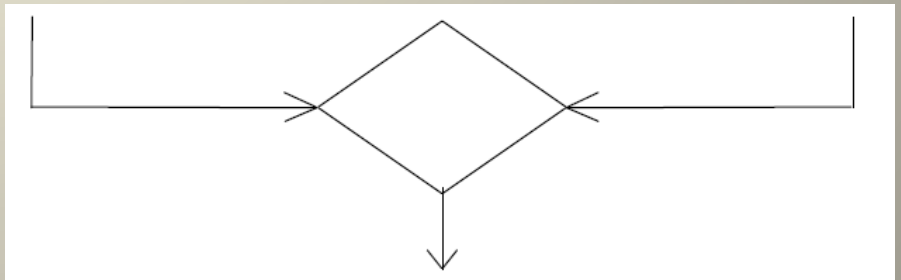
Multiple Branches: Examples

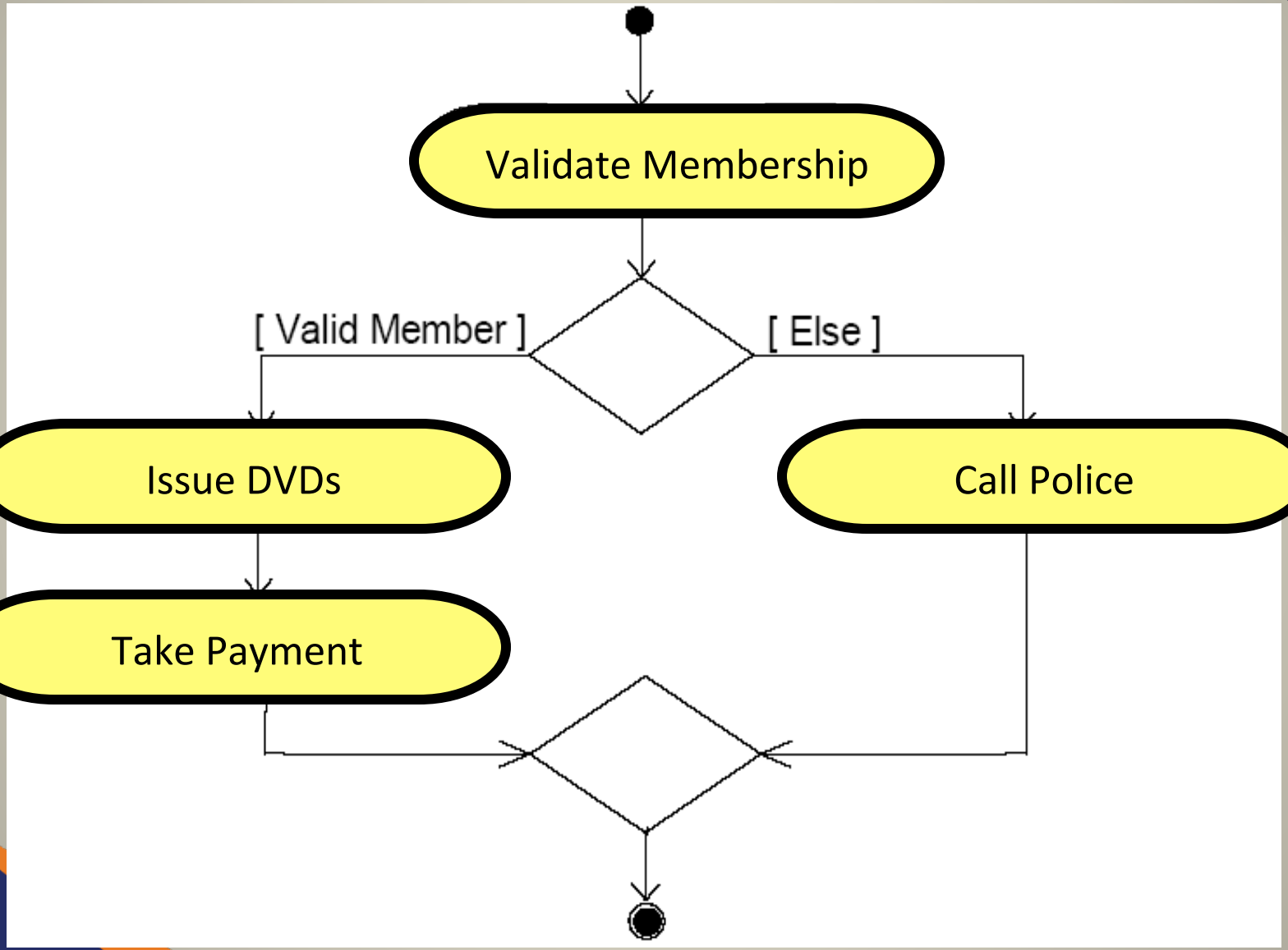


SLIIT
Discover Your Future

Merge Node

- A **merge point** is used to merge the flow of control from two or more branch points back together.
- The UML equivalent of **ENDIF** in pseudo code, or "}" in C/C++/Java.
- The diagram below shows a merge graphically – the **diamond is optional**:




Validate Membership
[Valid Member]
[Else]
Issue DVDs
Call Police
Take Payment

SLIIT
Discover Your Future

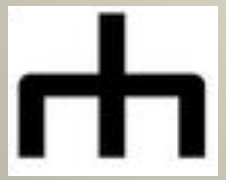
Activity - 2

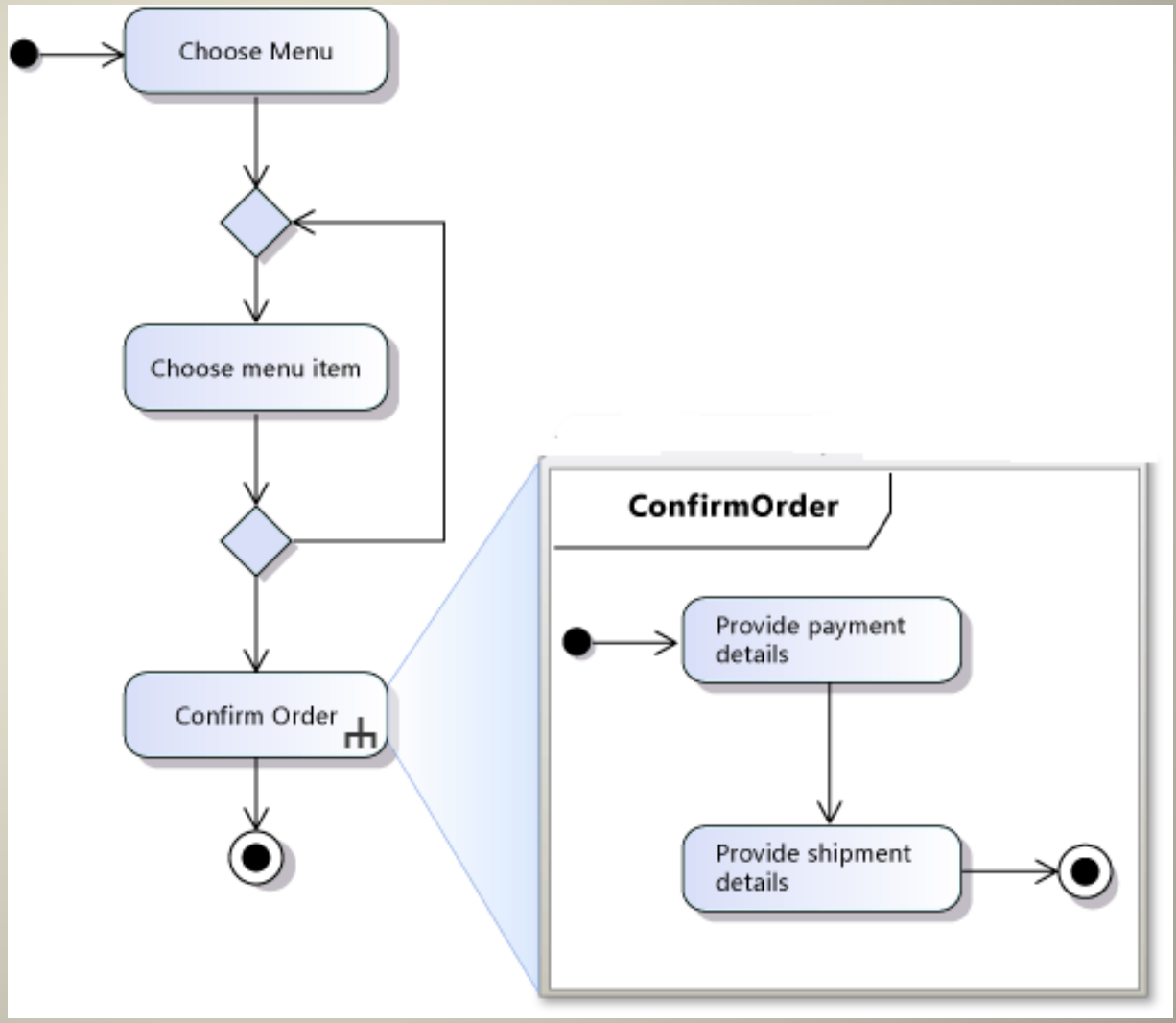
Draw an activity diagram for **"Borrow Books"** activity.

- To borrow books one must become a member.
- Members have access to core textbooks, reference books, general reading materials, CDs and DVDs.
- If a member student needs to borrow more than one book at a time, he/she can do so after depositing an additional refundable deposit of Rs. 3000/-

Call action /Sub Activity

- **Sub activity** is an activity that is defined in more detail on another activity diagram.
- It is indicated by a rake-style symbol within the action symbol.

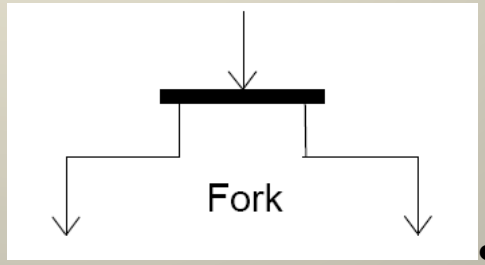




SLIIT
Discover Your Future

Forks & Joins

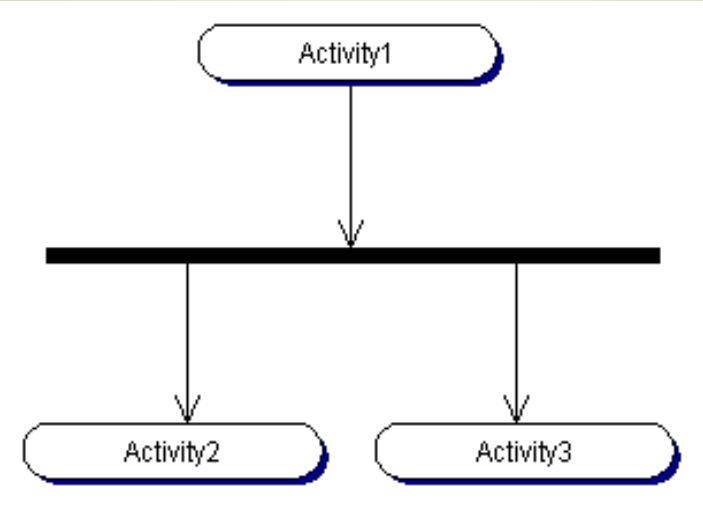
- Forks and joins are used to showing activities that can occur at the **same time** (in **parallel**).
  - this does not mean that the activities ***must*** occur concurrently in the finished software system.
  - it means that the **order of execution of the activities can be whatever** is convenient for the implementation.



SLIIT
Discover Your Future

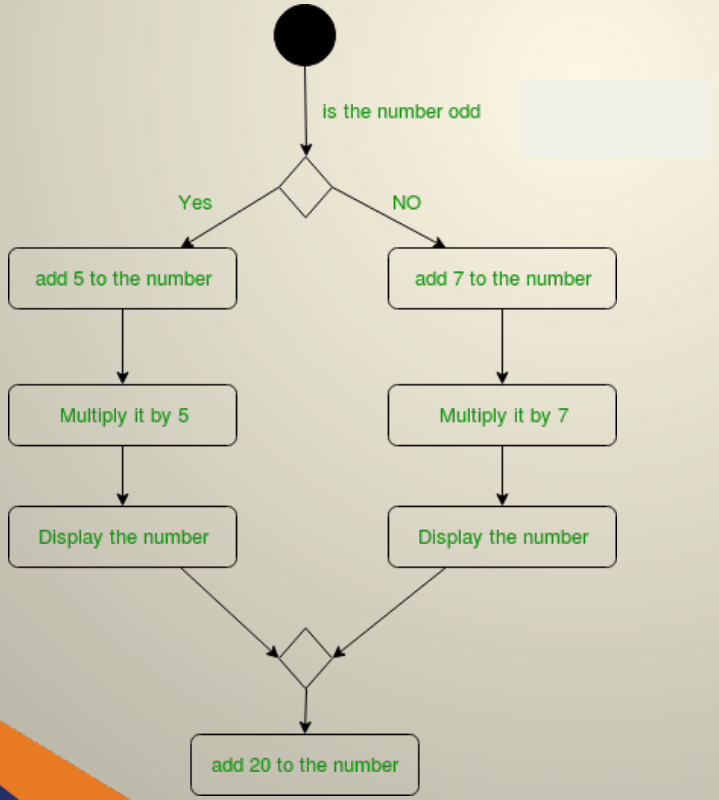
Fork

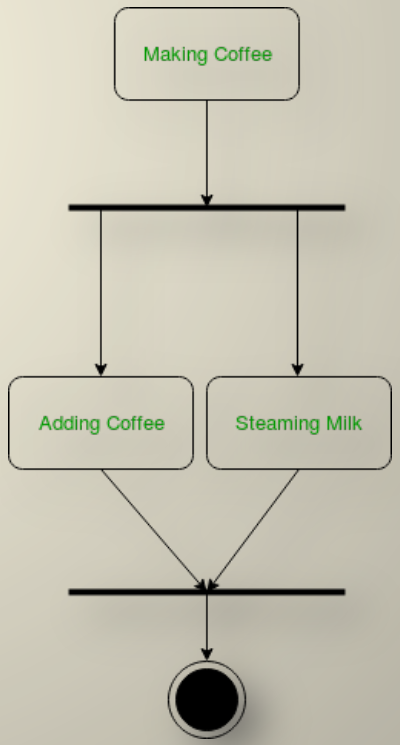
- A *fork* is when a single flow of control splits into two or more **parallel** (concurrent) flows of control.
- Represents a split in the flow of control.



Fork

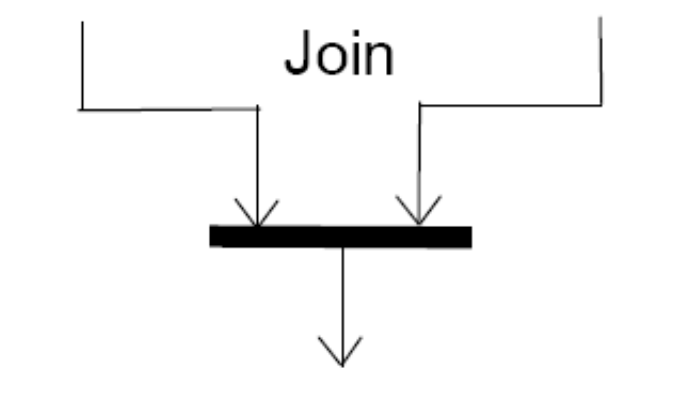
- Unlike a branch point, the **control flows down all forked paths**.
- With a branch point, the control flows down only **one** path.





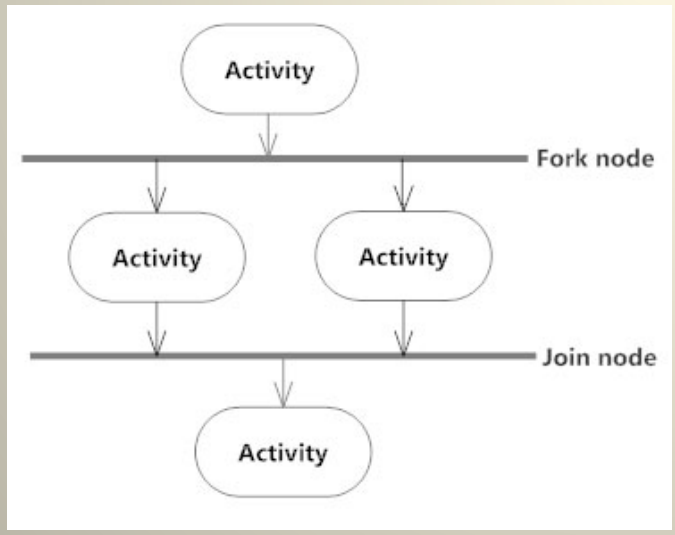
Join

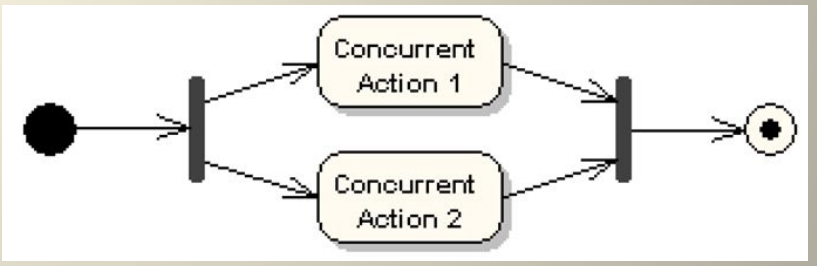
- A *join* is when **two or more flows of control merge into a single flow of control.**
- Represents the merging of multiple **flows** of control back into one.
- Every fork **must** have a join associated with it.



Join

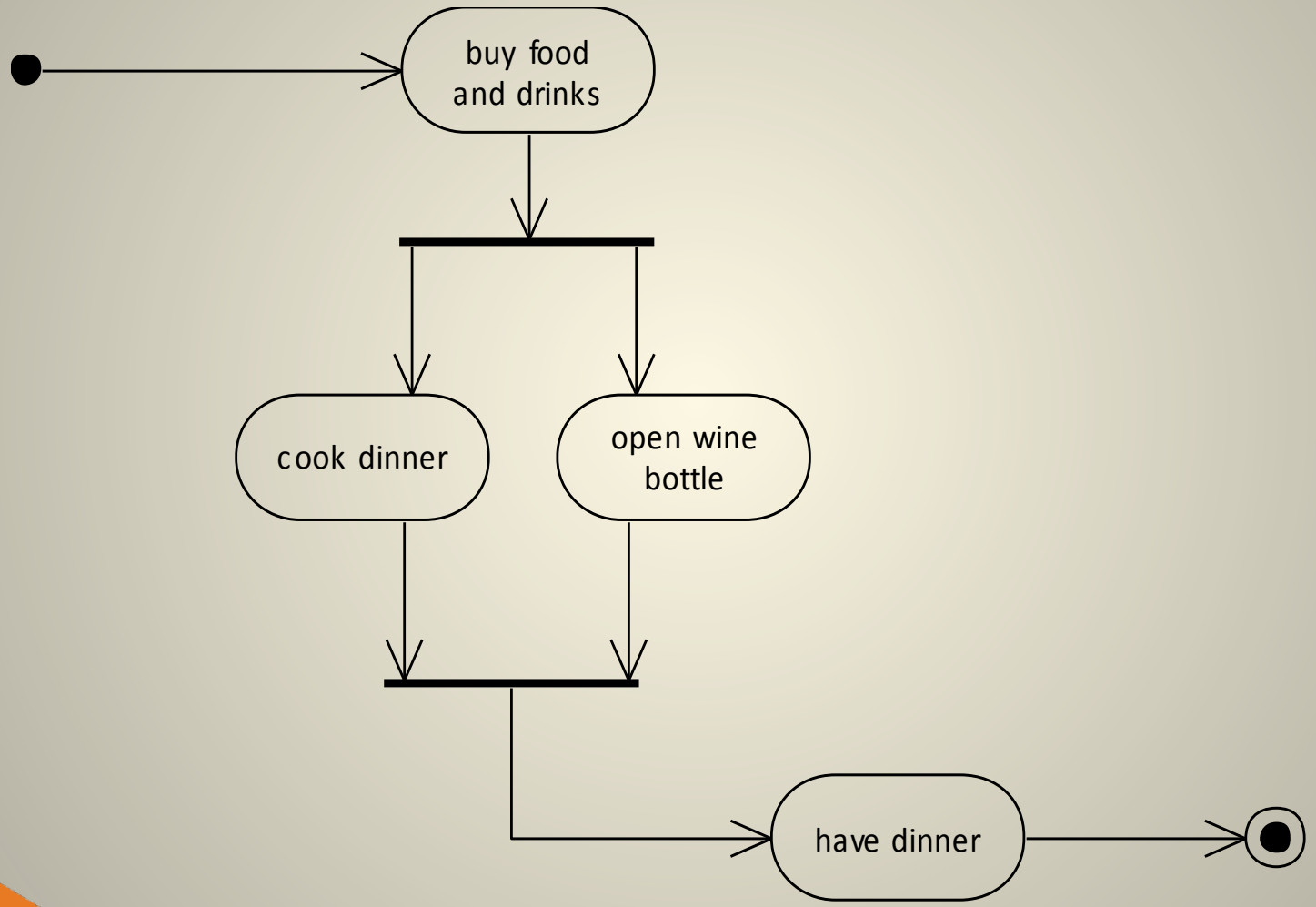
- **A flow of control** is also known as a **thread**.
- A **synchronization bar** is used to model forking and joining, and is modeled as a thick **horizontal or vertical bar.**





SLIIT
Discover Your Future

Fork and join example 1



SLIIT
Discover Your Future

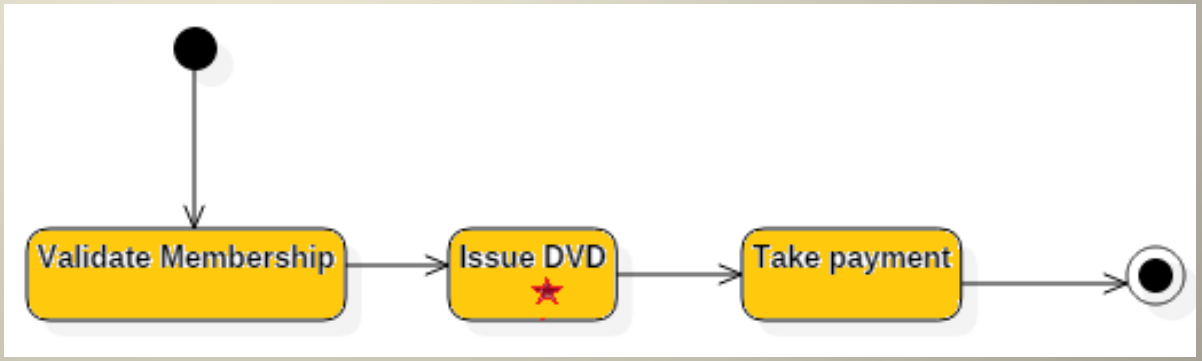
Activity - 3

"Enrolling in University"

1. Candidates who wish to register for university have to fill out enrolment forms and submit promptly.
2. The forms which are being incorrectly filled will be rejected.
3. The candidates who submitted properly filled forms are being enrolled.
4. The enrolled candidates then have to register for a seminar and make payments for initial tuitions.
5. Meanwhile they have to attend university overview presentation as a start.
6. Then only students can start classes.

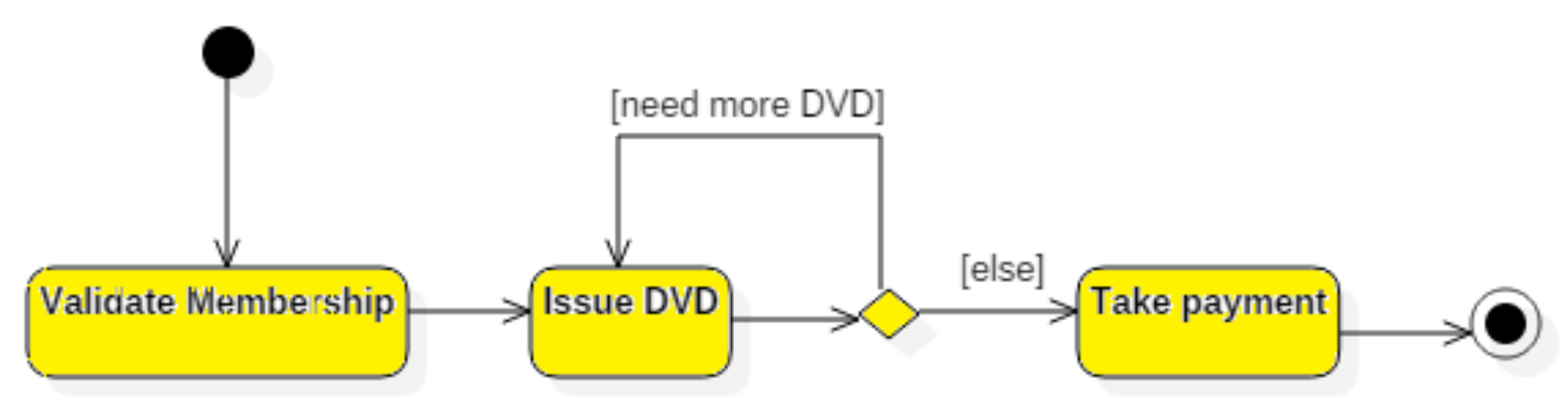
Iteration ★

- An **asterisk** inside an action state indicates that it may need to be **performed more than once.**
  - This notation is used to show iteration, without the unnecessary details of a loop construct.
- The next action state does not occur until the loop is finished.



Iteration

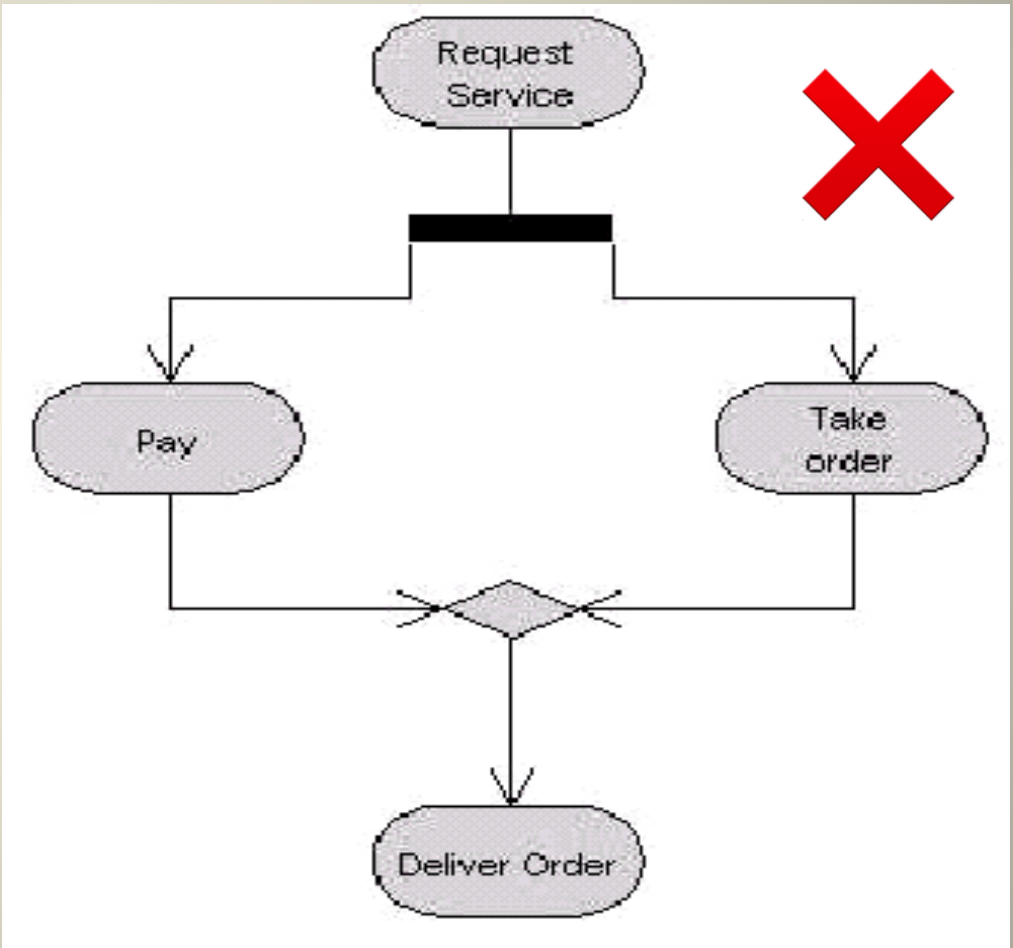
- Use of the **asterisk** to represent an iteration will not clearly highlight the **loop termination conditions** and **number of repetitions.**
- Therefore, use of a **Decision Node** to represent an iteration would be ideal.



SLIIT
Discover Your Future

Common Mistakes

- Incorrect use of **forks, branch points, merge points and joins.**
- The fork means that the flow of control goes down both paths.
- The merge point indicates a merging of divergent **paths, not flows**.

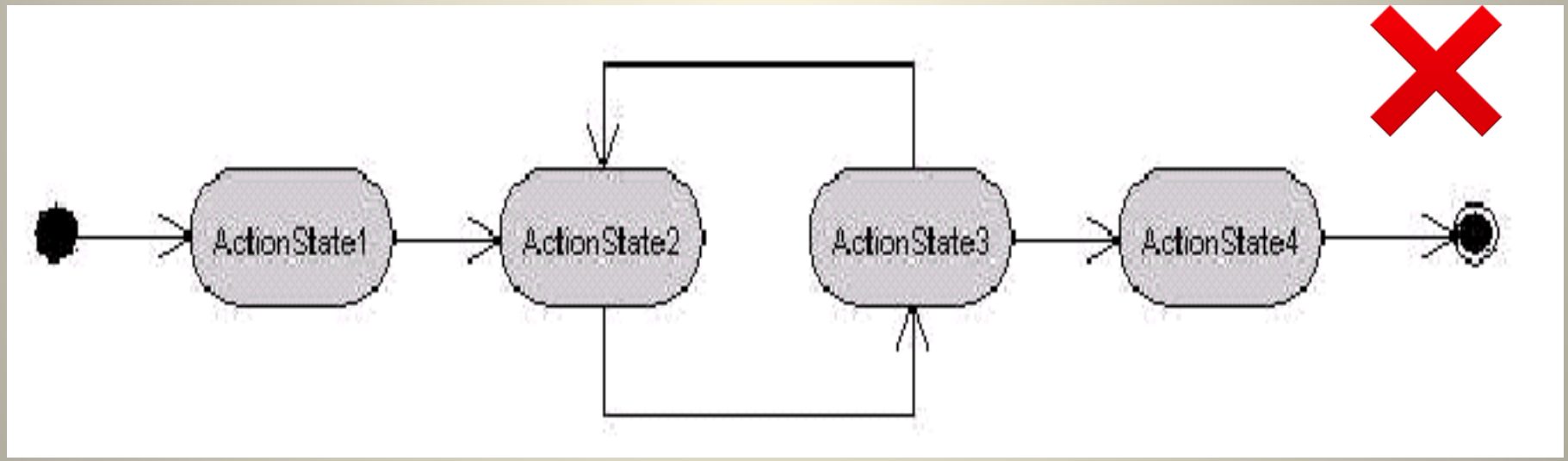


SLIIT
Discover Your Future

Common Mistakes

Loops.

- Guard conditions should be mentioned



SLIIT
Discover Your Future

Activity - 4

"Develop a software system for a client"

1. Project Manager (**PM**) first gathers the requirements
2. UI Engineer (**UIE**) develops the prototype
3. PM shows the prototype to **client**
4. Till the client is satisfied, UIE modifies the prototype.
5. Client signs off the prototype
6. Once the client is satisfied, UIE develop the UI screens, Software Engineer (**SE**) develops the system.
7. Once both (system and UI development) are done, SE integrates the system.
8. PM delivers the system to client
9. Client signs off the delivery.

Partitioning

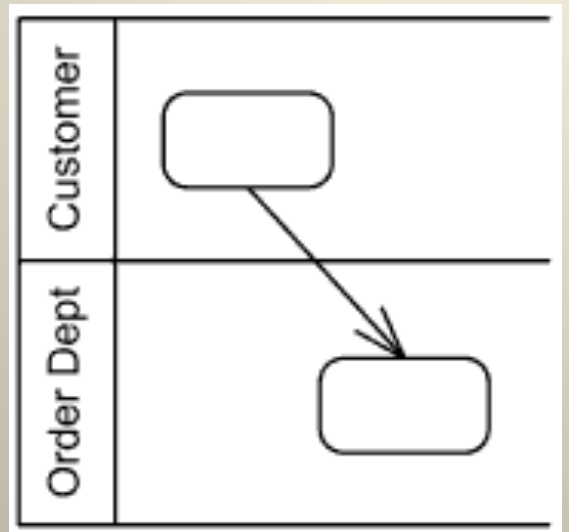
- An activity **partition** is an **activity group** for actions that have some common characteristic.
- Partitions often correspond to **organizational units** or **business actors** in a **business model**.

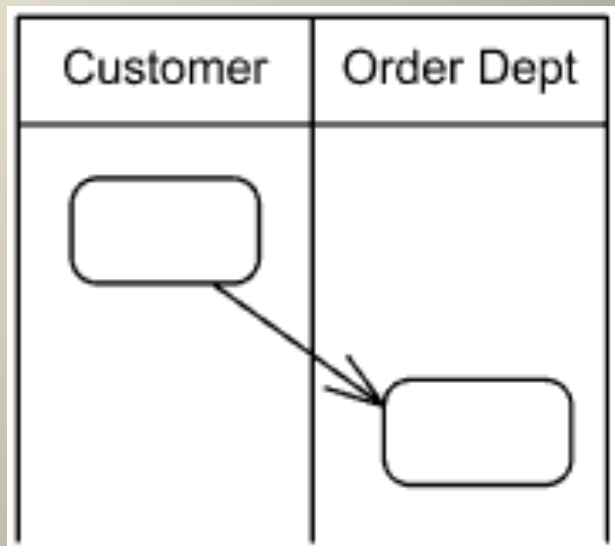
SLIIT
Discover Your Future

Partitions: SwimLanes

Activity **partition** may be shown using a **SwimLane** notation:

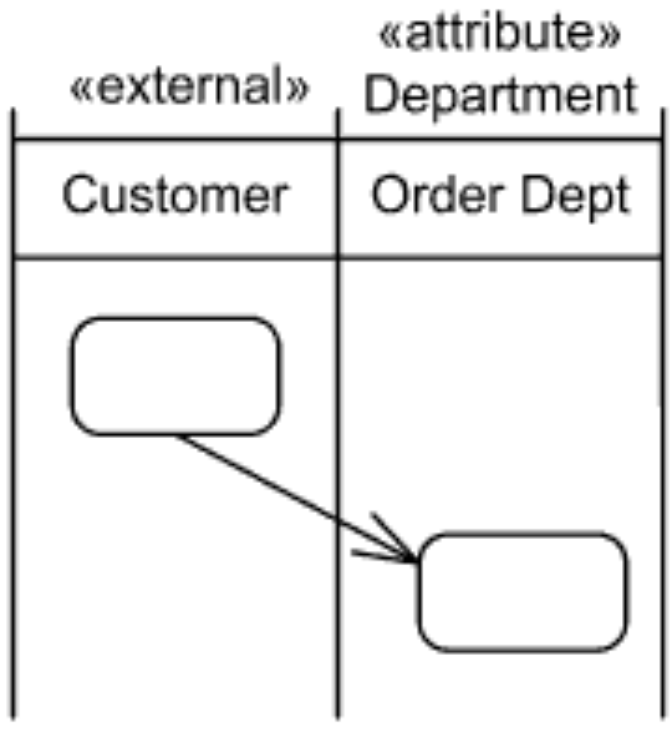
- with two, usually parallel lines, either horizontal or vertical
- a name labeling the partition in a box at one end.





Sub Partitioning in SwimLanes

- Order department is a sub class under Department class.



Activity - 5

"Develop a software system for a client"

1. Project Manager (**PM**) first gathers the requirements.
2. UI Engineer (**UIE**) develops the prototype.
3. PM shows the prototype to **client.**
4. Till the client is satisfied, UIE modifies the prototype.
5. Once the client is satisfied, signs off the prototype
6. UIE develop the UI screens while Software Engineer (**SE**) develops the system.
7. Once both (system and UI development) are done, SE integrates the system.
8. PM delivers the system to client
9. Client signs off the delivery.

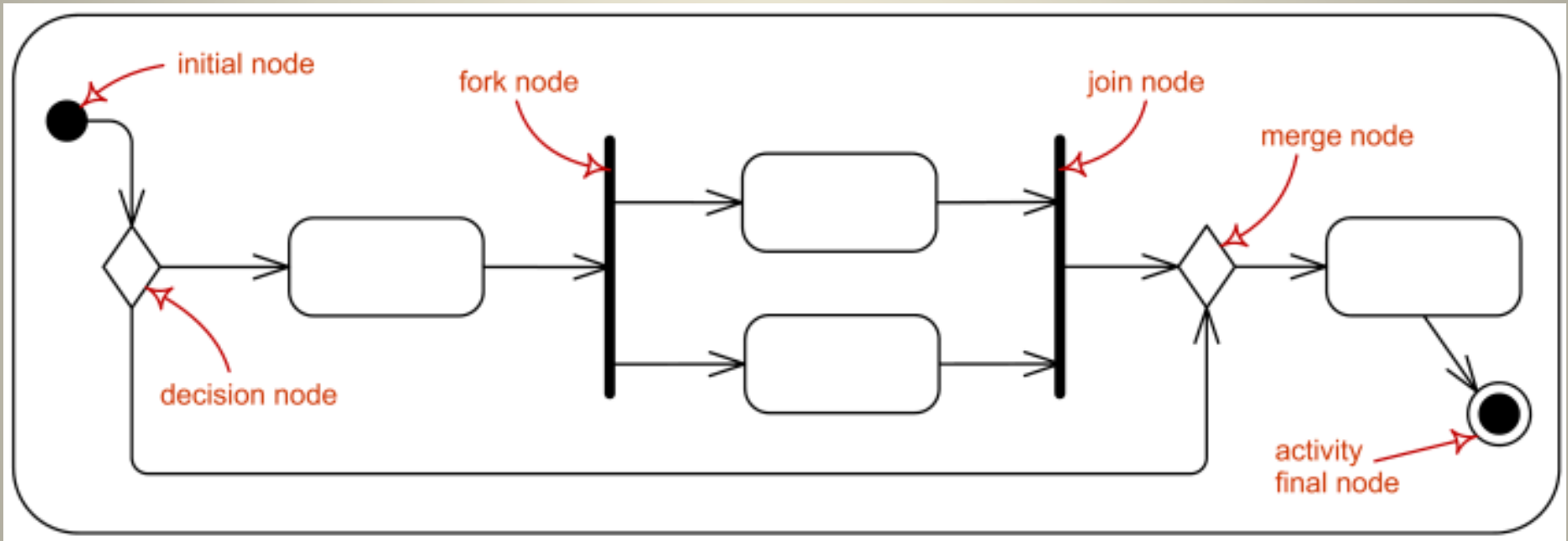
SLIIT
Discover Your Future

Activity - 6
"Order Processing System"

1. Once the order is received by the customer service department, several other departments too operate on various activities to complete the order.
2. The fulfillment department should fill the order (prepare goods to deliver) and deliver the order to customer.
3. Meanwhile, as soon as the order is received, customer service department works on sending an invoice to the customer.
4. The finance department will handle the payments from the customer. The shop operates on credit basis. Therefore, receiving payments is not mandatory prior to the delivery of the goods.
5. Customer service department can close the order only after completing all the above activities.

SLIIT
Discover Your Future

Activity Diagram Notations



SLIIT
Discover Your Future

How to create an Activity Diagram

1. Identify activities (steps) of a process
2. Identify who/what performs activities (process steps)
3. Draw swim lanes
4. Identify decision points (if-then)
5. Determine if a step is in loop (*For each…, or* if-then loop)
6. Determine if step is parallel with some other
7. Identify order of activities, decision points

Continues…

Reference - Systems Analysis & Design * Bob Travica

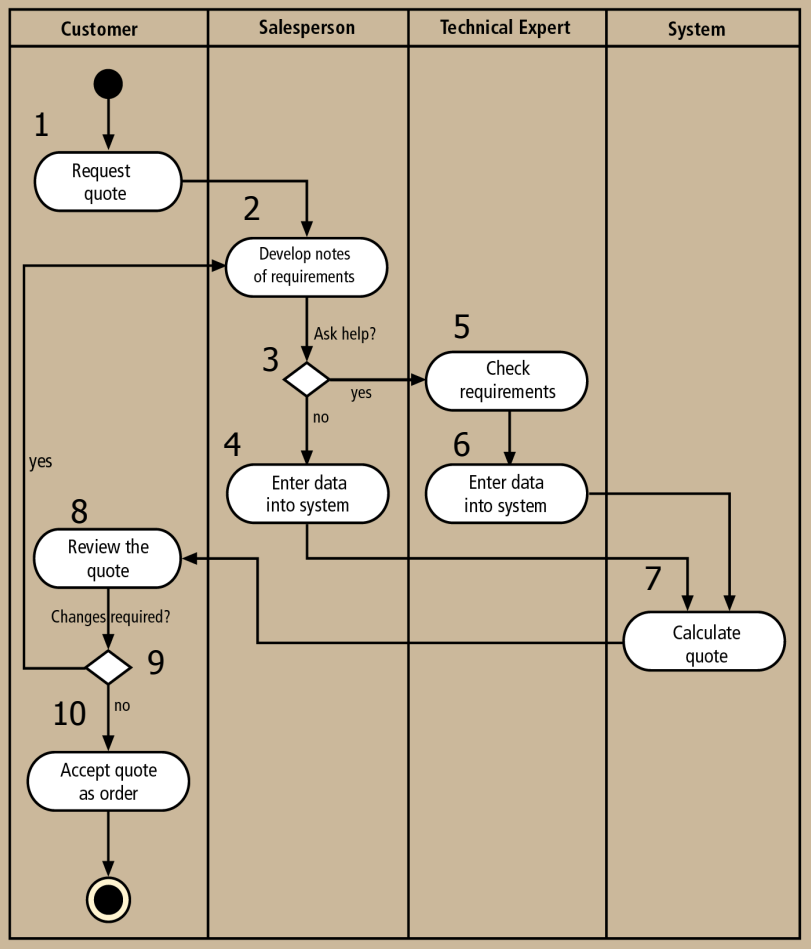
How to create an Activity Diagram (cont.)

8. Draw the start point of the process in the swimlane of the first activity (step)
9. Draw the oval of the first activity (step)
10. Draw an arrow to the location of the second step
11. Draw subsequent steps, while inserting decision points and synchronization/loop bars where appropriate
12. Draw the end point after the last step.

You can tabulate these data (see next slide).

Reference - Systems Analysis & Design * Bob Travica

How to create an Activity Diagram (cont.)



Step ID	Process Activity or Decision	Who/ What Performs	Parallel Activity	Loop	Preceding Step
1	Request quote	Customer	No	No	-
2	Develop requirement notes	Salesperson	No	Yes	1
3	Decision: Help?	Salesperson	-	Yes	2
4	Salesperson enters data	Salesperson	No	Yes	3
5	Check requirements	Technical Expert	No	Yes	3
6	Tech. expert enters data	Technical Expert	No	Yes	5
7	Calculate quote	System	No	Yes	4, 6
8	Review quote	Customer	No	Yes	7
9	Decision: Changes?	Customer	No	Yes	8
10	Accept quote as order	Customer	No	No	9

Reference - Systems Analysis & Design * Bob Travica

SLIIT
Discover Your Future

References

- The Unified Modeling Language Reference Manual – James Rumbaugh, Ivar Jacobson, et al
- UML Distilled: A Brief Guide to the Standard Object Modeling Language by Martin Fowler, Kendall Scott
- Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process by Craig Larman
- The Complete UML Training Course, Student Edition by Grady Booch, et al
- UML Explained by Kendall Scott

SLIIT
Discover Your Future

More Examples .....

SLIIT
Discover Your Future

Use Case for Order Processing

The following scenario explains how an order is processed.

1. When the Order Processing Depatment (OPD) receives an order, they have to check whether the order is in proper order as given.
2. OPD checks each line item on the order to see whether they have the goods in stock. If they do, they assign the goods to the order.
3. After asigning, if the goods are not in stock or if the stock level of those asigned goods goes below the reorder level, OPD will reorder the goods.

Use Case for Order Processing

4. While the OPD is checking the order, Finance Department (FD) checks to see if the payment is OK.
5. Once the payment is OK and OPD has assigned the goods OPD will check whether any items are sent for reorder. If there are items to reorder, OPD will leave the order waiting.
6. Otherwise OPD will dispatch the order. If the payment isn't OK, OPD will cancel the order."