



SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107

Accredited by NAAC-UGC with 'A' Grade

**Approved by AICTE, Recognized by UGC & Affiliated to Anna University,
Chennai**

Department of Artificial Intelligence and Data Science

Object Oriented Software Engineering

Unified Modeling Language (UML) Activity Diagrams

**Prepared By
R.Sowmiya,AP/AI&DS
SNSCE.**

Activity Diagrams

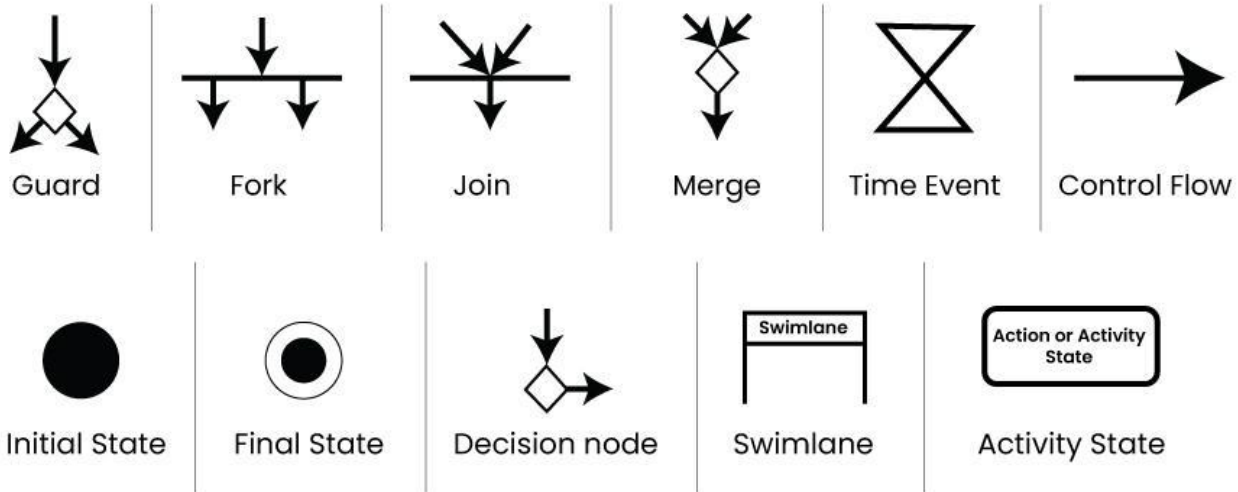
- Activity diagrams show the steps involved in how a system works, helping us understand the flow of control.
- They display the order in which activities happen and whether they occur one after the other (sequential) or at the same time (concurrent).
- These diagrams help explain what triggers certain actions or events in a system.
- An activity diagram starts from an initial point and ends at a final point, showing different decision paths along the way.
- They are often used in business and process modeling to show how a system behaves over time.



When to use Activity Diagram?

- Activity diagrams are useful in several scenarios, especially when you need to visually represent the flow of processes or behaviors in a system. Here are key situations when you should use an activity diagram:
- **Modeling Workflows or Processes:** When you need to map out a business process, workflow, or the steps involved in a use case, activity diagrams help visualize the flow of activities.
- **Concurrent or Parallel Processing:** If your system or process involves activities happening simultaneously, an activity diagram can clearly show the parallel flow of tasks.
- **Understanding the Dynamic Behavior:** When it's essential to depict how a system changes over time and moves between different states based on events or conditions, activity diagrams are effective.
- **Clarifying Complex Logic:** Use an activity diagram to simplify complex decision-making processes with branching paths and different outcomes.
- **System Design and Analysis:** During the design phase of a software system, activity diagrams help developers and stakeholders understand how different parts of the system interact dynamically.
- **Describing Use Cases:** They are useful for illustrating the flow of control within a use case, showing how various components of the system interact during its execution.

Activity Diagram Notations



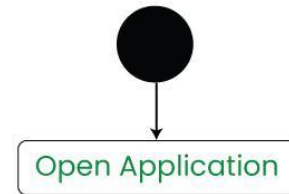
1. Initial State

The starting state before an activity takes place is depicted using the initial state.



Initial State

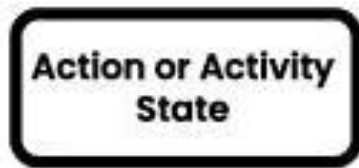
Initial State symbol being used



Unified Modeling Language (UML) | Activity Diagrams

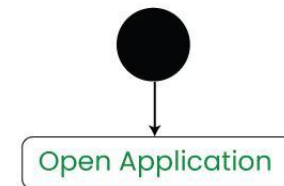
2. Action or Activity State

An activity represents execution of an action on objects or by objects. We represent an activity using a rectangle with rounded corners. Basically any action or event that takes place is represented using an activity.



Activity State

Activity State symbol being used



Unified Modeling Language (UML) | Activity Diagrams

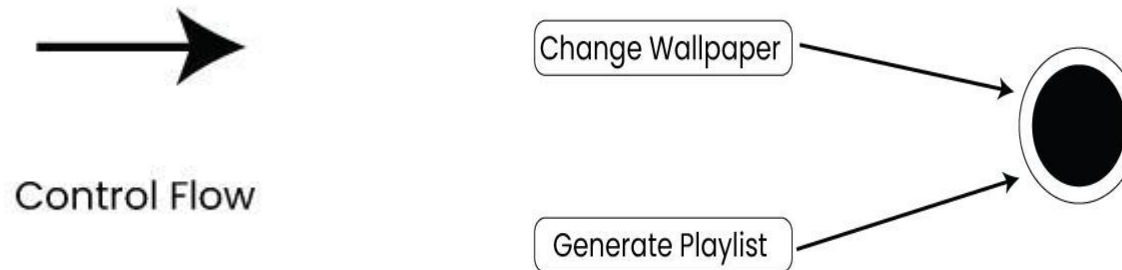
3. Action Flow or Control flows

Action flows or Control flows are also referred to as paths and edges.

They are used to show the transition from one activity state to another activity state.

Here both the states transit into one final state using action flow symbols i.e. arrows.

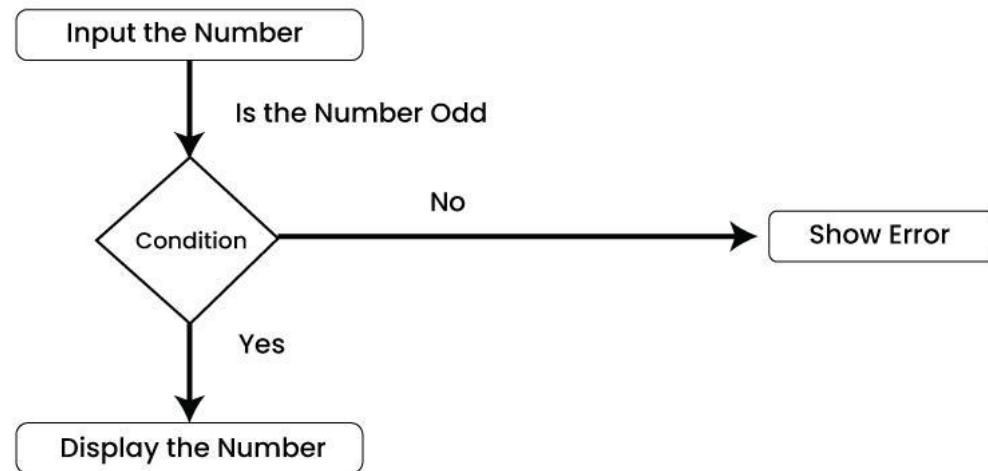
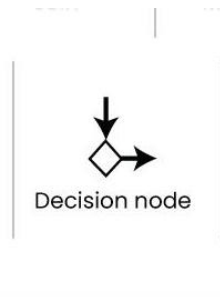
Using Action Flows for Transitions



4. Decision node and Branching

When we need to make a decision before deciding the flow of control, we use the decision node. The outgoing arrows from the decision node can be labelled with conditions or guard expressions. It always includes two or more output arrows.

An Activity Diagram using Decision Node



Unified Modeling Language (UML) | Activity Diagrams

5. Guard

A Guard refers to a statement written next to a decision node on an arrow sometimes within square brackets.

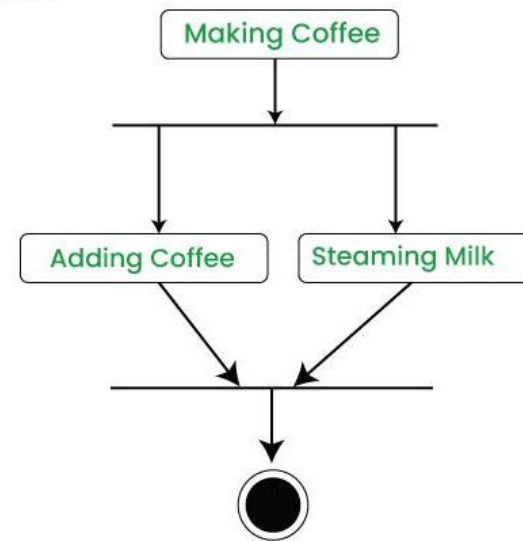


7. Join

Join nodes are used to support concurrent activities converging into one. For join notations we have two or more incoming edges and one outgoing edge.



A Diagram using Join Notation



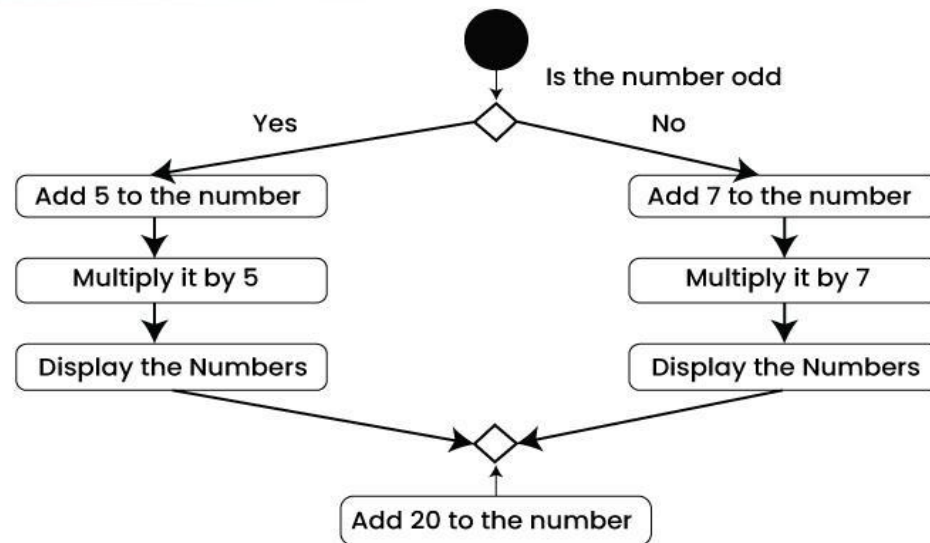
Unified Modeling Language (UML) | Activity Diagrams

8. Merge or Merge Event

Scenarios arise when activities which are not being executed concurrently have to be merged. We use the merge notation for such scenarios. We can merge two or more activities into one if the control proceeds onto the next activity irrespective of the path chosen.



An Activity Diagram using Merge Notation

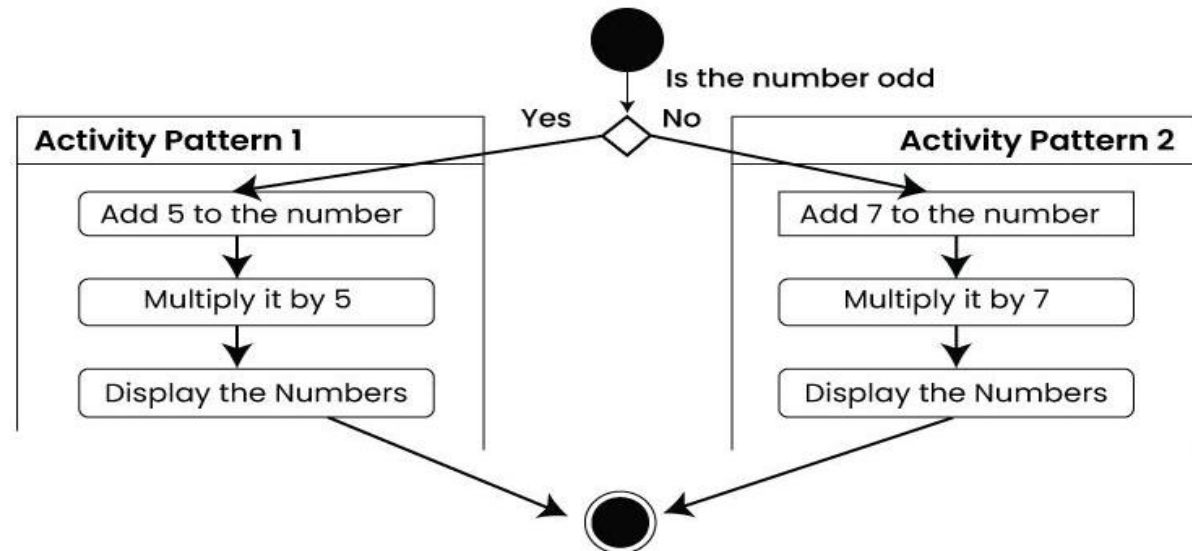
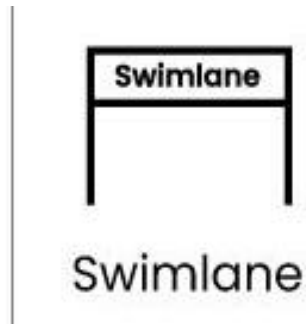


Unified Modeling Language (UML) | Activity Diagrams

9. Swimlanes

We use Swimlanes for grouping related activities in one column. Swimlanes group related activities into one column or one row. Swimlanes can be vertical and horizontal. Swimlanes are used to add modularity to the activity diagram. It is not mandatory to use swimlanes. They usually give more clarity to the activity diagram. It's similar to creating a function in a program. It's not mandatory to do so, but, it is a recommended practice.

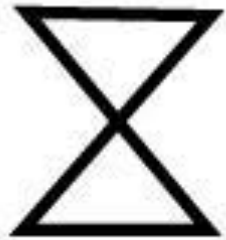
An Activity Diagram making use of Swimlanes



Unified Modeling Language (UML) | Activity Diagrams

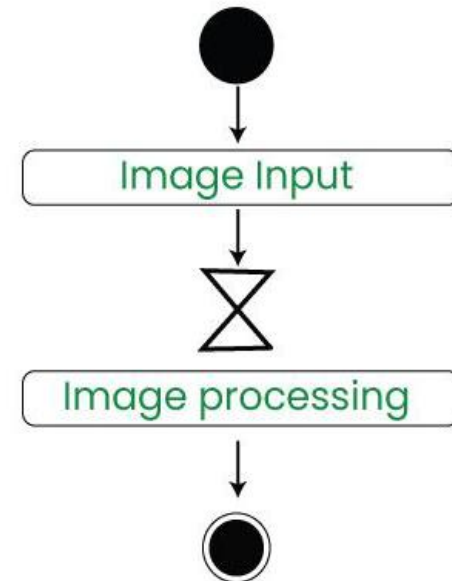
10. Time Event

This refers to an event that stops the flow for a time; an hourglass depicts it. We can have a scenario where an event takes some time to completed.



Time Event

An Activity Diagram using Time Event Notation



Unified Modeling Language (UML) | Activity Diagrams

11. Final State or End State

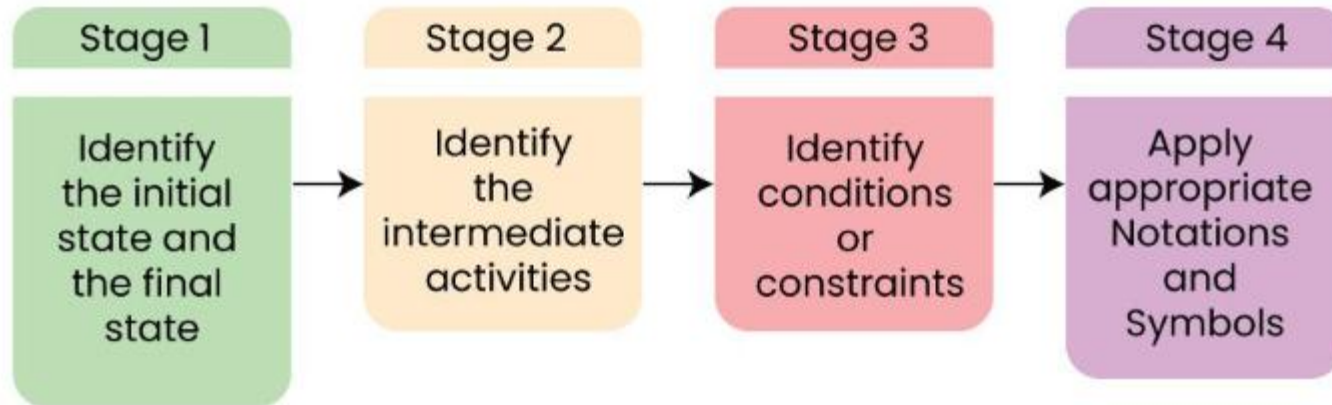
- The state which the system reaches when a particular process or activity ends is known as a Final State or End State.
- We use a filled circle within a circle notation to represent the final state in a state machine diagram.
- A system or a process can have multiple final states.



Final State

How to Draw an Activity Diagram in UML?

Steps to Draw an Activity Diagram



Example of Activity diagram for library management System

