

UML Notations

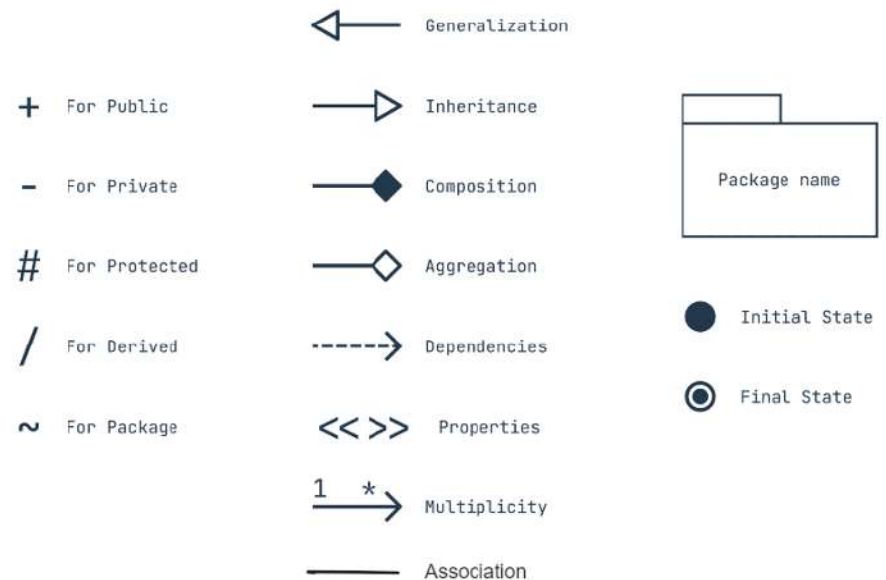
UML is composed of three main building blocks:

- Things
- Relationships
- Diagrams

These three exist at the **CENTER of UML** and play a key role in producing effective and easily understandable models.

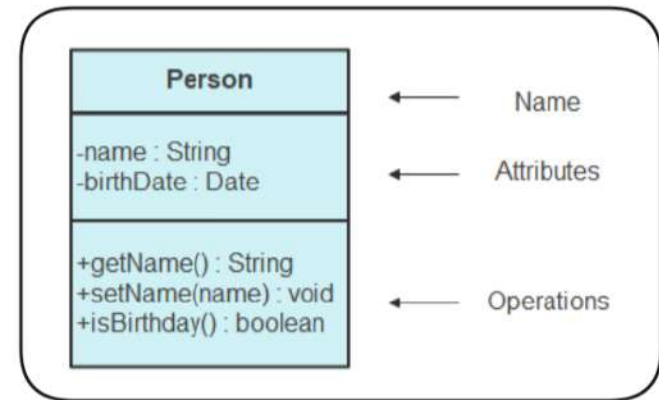
1. Class Notation
2. Object Notation
3. Interface Notation
4. Use Case Notation
5. Actor Notation
6. Component Notation
7. Node Notation
8. Activity Diagrams
9. Interaction Diagrams

UML NOTATIONS



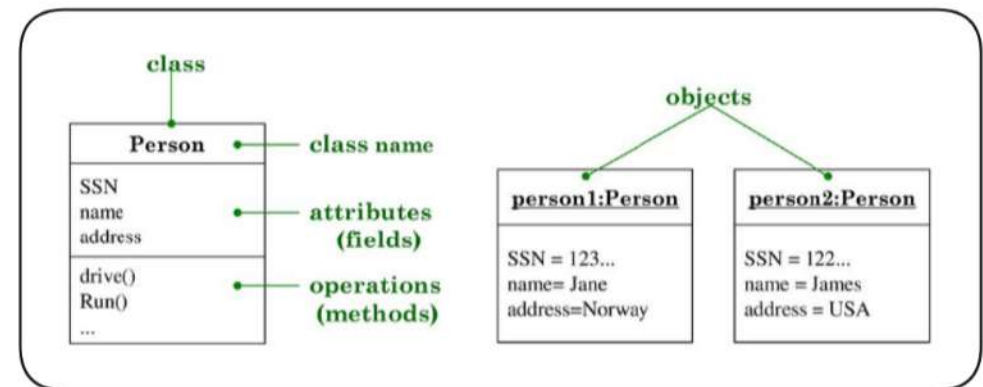
Class Notation

Class notation represents the **attributes** and **methods** of an object.



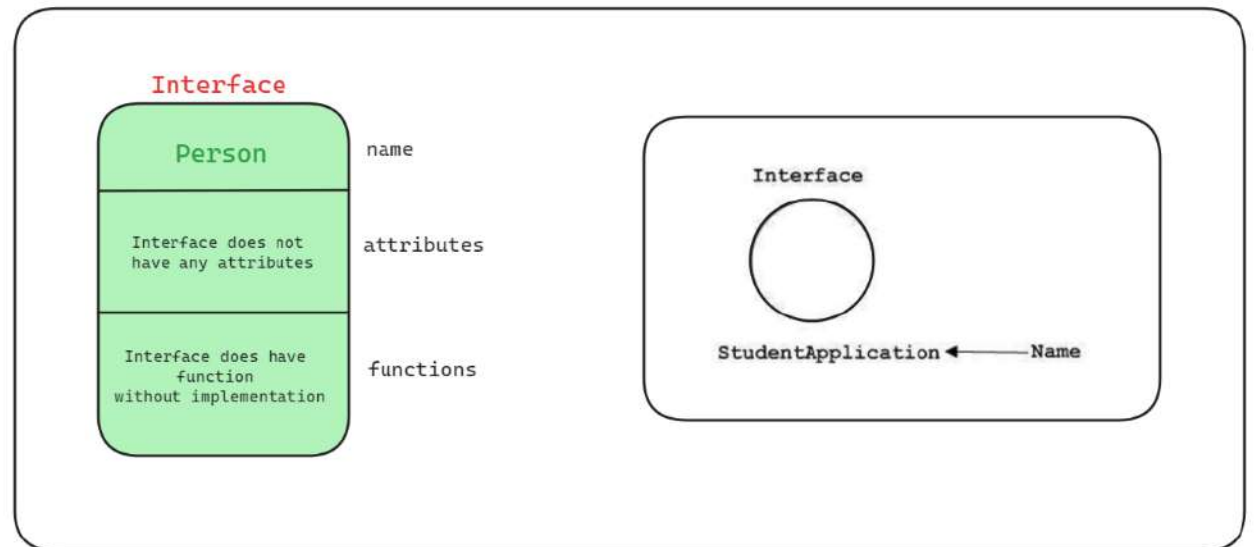
Object Notation

Object notation refers to the *instance* of a class.



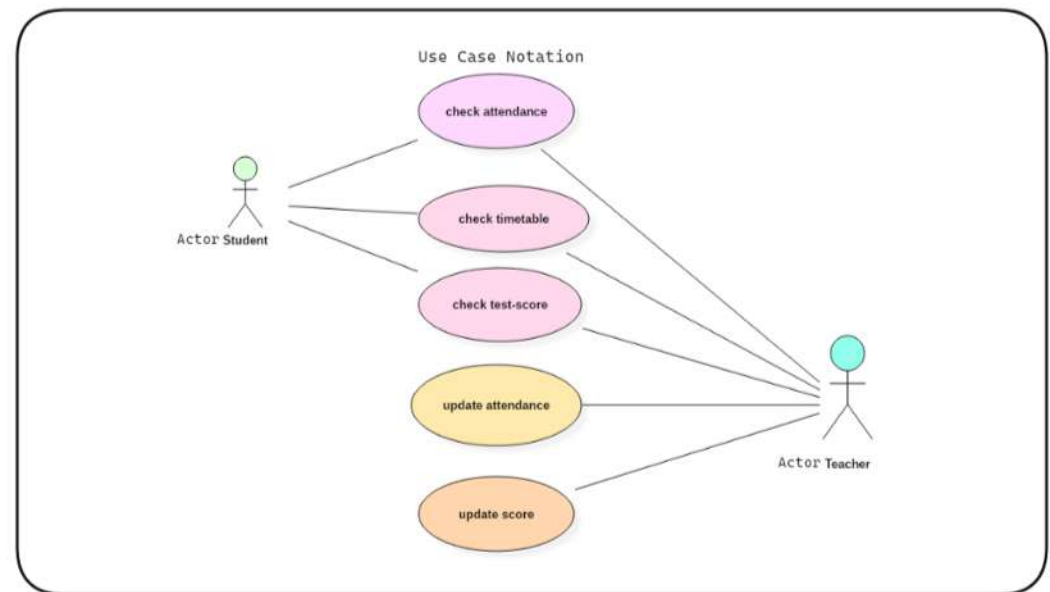
Interface Notation

Interface notation represents the functionality *without its implementation*.



Use Case Notation

Use Case notation describes the user's **goals** and possible **interactions** with the system.



Actor Notation

Actor notation represents the **entities** interacting with system.



Primary Actor:

Left

Definition: The primary actor is a user or system that directly interacts with the system to achieve a specific goal or perform a use case.

Role: Primary actors are essential for the success of a use case and are typically the main beneficiaries or initiators of a particular system functionality.

Representation: In a use case diagram, primary actors are depicted as stick figures or blocks outside the system boundary, connected to the use cases they initiate.

Secondary Actor:

Right

Definition: The secondary actor is a user or system that provides a supporting role or assists in achieving a use case's goals but is not the primary actor for that use case.

Role: Secondary actors are not the main initiators or beneficiaries of a specific use case but may still have an influence or participate in the use case's execution.

Representation: In a use case diagram, secondary actors are also depicted as stick figures or blocks outside the system boundary, connected to the use cases they support.

Example 1:

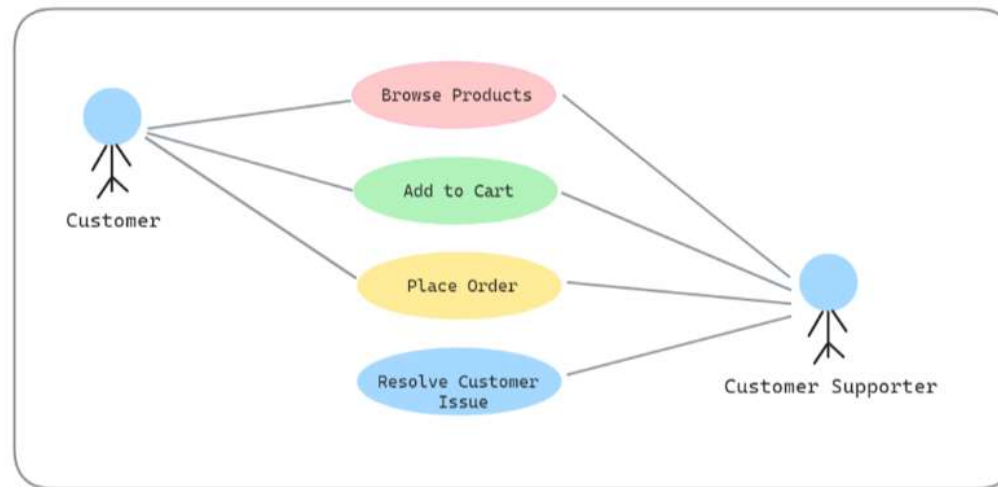
Consider an online shopping system. In a use case diagram:

Primary Actor: Customer

The customer is the primary actor initiating use cases like "**Browse Products**," "**Add to Cart**," and "**Place Order**."

Secondary Actor: Customer Support

Customer Support may be a secondary actor, providing assistance or support in the use case "**Resolve Customer Issues**" but is not the main initiator of that use case.



Example 2:

Let's consider the roles of students and teachers in the context of a hypothetical educational system.

Primary Actor: Student

Students are primary actors in an educational system. They directly interact with the system to achieve various educational goals and perform specific tasks: **Enrolling in courses, Accessing course materials, Submitting assignments, Taking exams, Checking grades**

Primary Actor: Teacher

Teachers or instructors are primary actors who interact with the educational system to manage courses, assess student performance, and facilitate the learning process: **Creating and managing courses, Uploading course materials, Evaluating assignments and exams, Providing feedback to students, Managing grades and records**

Secondary Actor: Administration

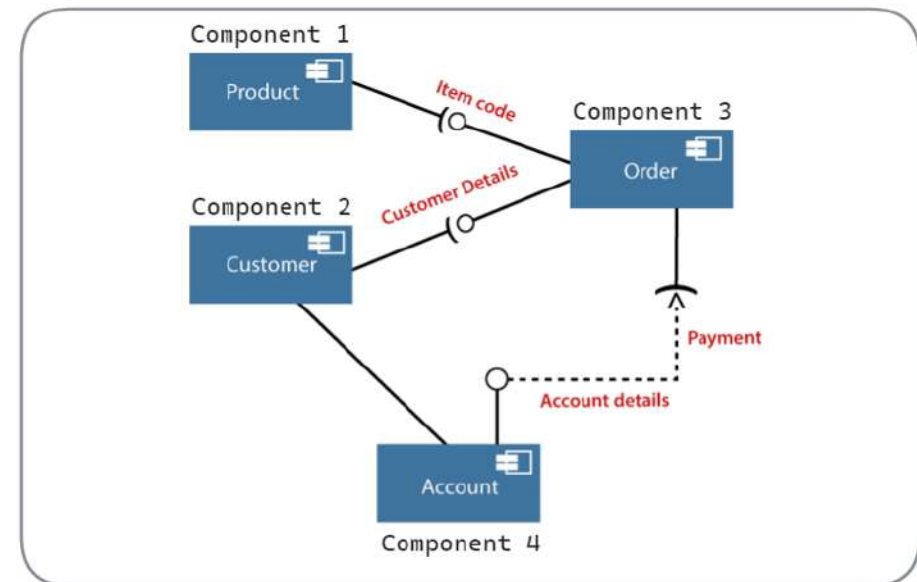
The administration may serve as a secondary actor, supporting both students and teachers in administrative tasks related to the educational system: **Managing student enrollment, Handling course scheduling, Administering system updates, Providing support for technical issues**

Secondary Actor: Parent/Guardian

Parents or guardians may act as secondary actors, particularly in systems designed to provide information or updates to them regarding their child's educational progress: **Accessing student grades and reports, Receiving communication from the educational system**

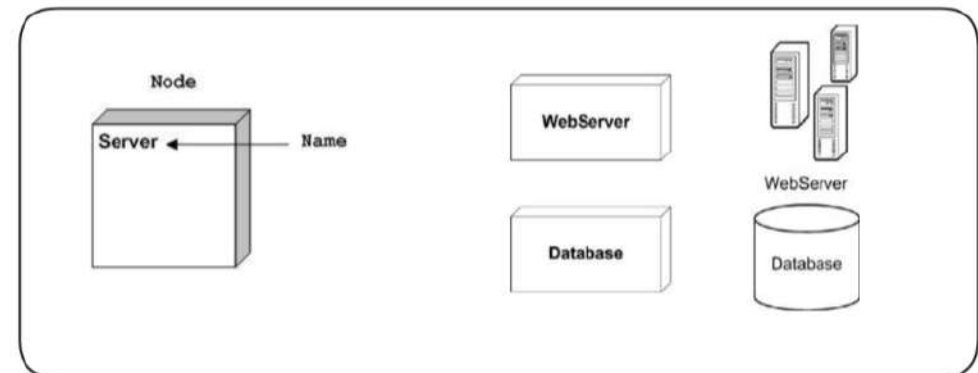
Component Notation

Component notation represents the **section** of the system.



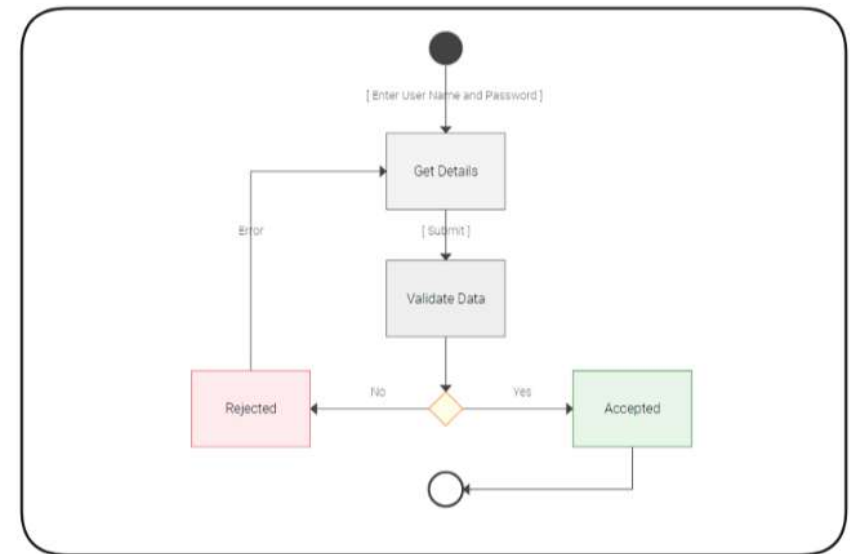
Node Notation

Node notation is similar to the component notation, with the difference being that the node notation refers to the **physical aspect (Hardware)** of a system, such as a server.



Activity Diagrams

Activity Diagrams describe the various **interactions** performed by different **components** present in the system.



Activity Diagram is like a Flow Chart Diagram

Interaction Diagrams

Interaction Diagrams describes the *message flow* (*How Execution Flow*) between the different components present in the system.

