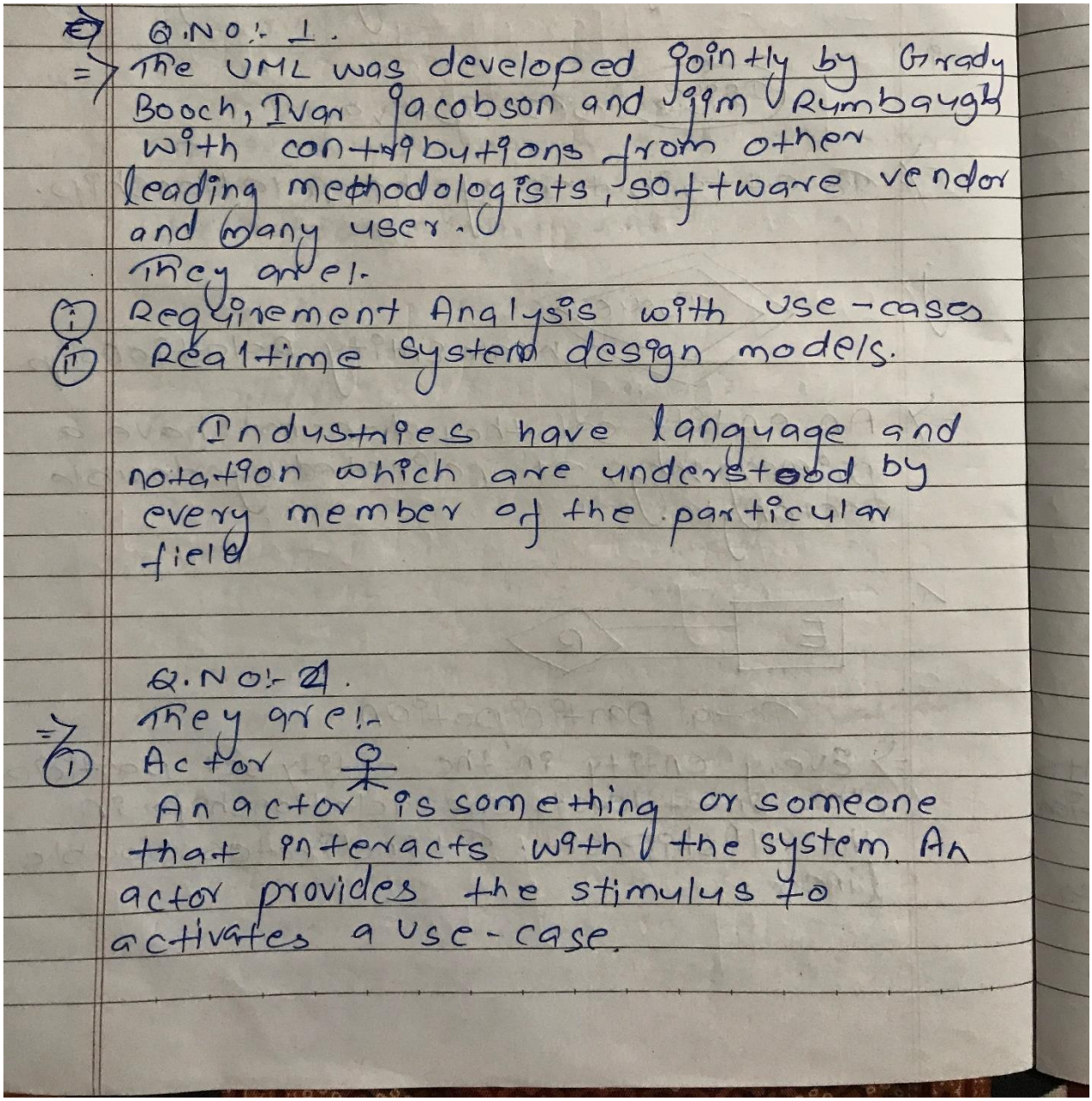
Workshop: 4

2332917_NirajChaudhary_L4CG16



Q.NO:- 1.

=> The UML was developed jointly by Grady Booch, Ivar Jacobson and Jim Rumbaugh with contributions from other leading methodologists, software vendor and many user.

They are:-

(i) Requirement Analysis with use-cases

(ii) Realtime system design models.

Industries have language and notation which are understood by every member of the particular field

Q.NO:- 2.

=> They are:-

(i) Actor

An actor is something or someone that interacts with the system. An actor provides the stimulus to activates a use-case.

modeled

2. system^ [system]

=> The process of developing abstract models of a system, with each model presenting a different view or perspective of that system.

3 use-cases [use case]

=> A use-cases in UML is defined as a set of sequence of actions a system performs that yield an observable result of value to a particular actor.

4. Association

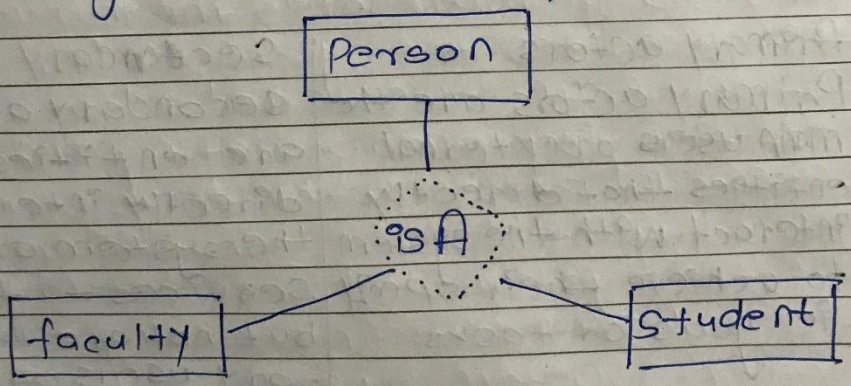
=> A line connecting an actor to a use case represents an association. It represents the fact that the actor communicates with the use case.

5 Dependency.

=> It relnship in which one element the client uses or depends on element the supplier. You can use dependency rlnship in class diagram, component diagram, deployment diagram and use-case diagram to indicate that a change to the supplier might require a change to the client.

The generalization rlnship is incorporated to record attributes, operation and rlnship in a parent model element so that it can inherited in one or more child model element.

The parent model element can be have as many child can have one or more parents. But most commonly, it can be seen that there is one parent model element and multiple child model elements. The generation rln-ship does not consists of names. The generation rlnship is represent




2=> It defines the software engineering, a use case is a list of actions or event steps, typically defining the interaction betn a role (known in the unified Modeling Language as an actor) and a system, to achieve a goal.

before actual coding begins, aiding in betters planning and decision making. It enables stakeholder to have a shared understanding of the system's structure, behavior and interactions, fostering teamwork and alignment of project goals. Additionally, UML's modular nature enhance reusability and maintainability, as components and relationships can be easily identified and managed, leading to higher quality software development.