Object-Oriented System Design - Assignment

Utkarsh Gupta
A2305217557
7CSE-8Y

Q1. Distinguish between method and message in object.

A message is a name for a responsibility which an object may have. A method is a named, concrete piece of code that encodes one way a responsibility may be fulfilled. You might say that it is one method by which a message might be implemented.

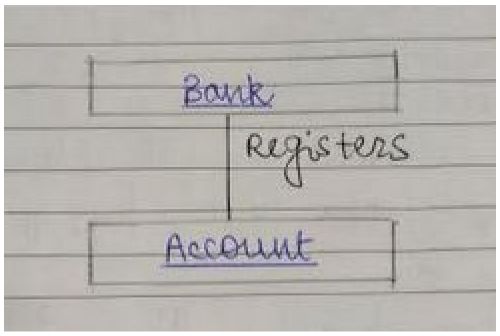
Q2. What is the need of modeling?

The Unified Modeling Language is a graphical language for Object Oriented Analysis Design that gives a standard way to write a software system's blueprint. It helps to visualize, specify, construct, and document the artifacts of an object-oriented system.

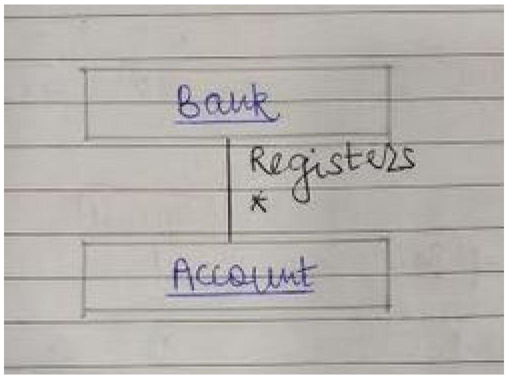
Q3. List various relationships exist in class diagram along with example.

The types of relationships that exist between classes, along with their notation, and also what they actually mean.

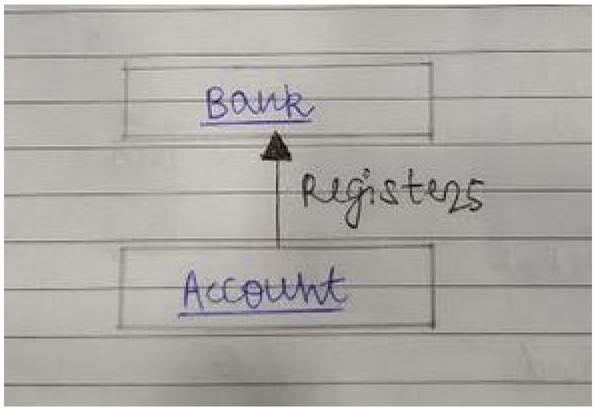
1. **Association**
An association relation is established when two classes are connected to each other in any way.



2. **Multiplicity**
An example of this kind of association is many accounts being registered by the bank. Hence, the relationship shows a star sign near the account class (one to many and many to many etc).

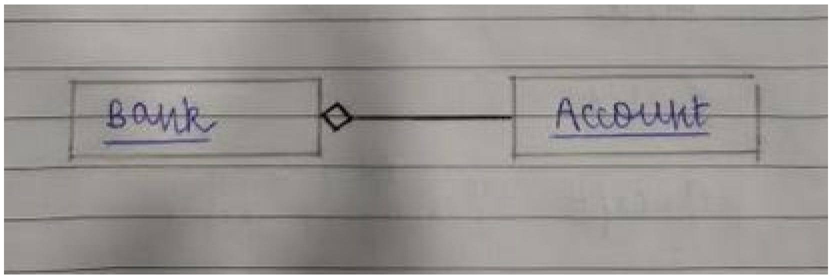


3. **Directed Association**
By default, an association that exists between classes is bi-directional. Ideally, you may illustrate the flow of the association by utilizing a directed association. The arrowhead indicates the container-contained relationship.

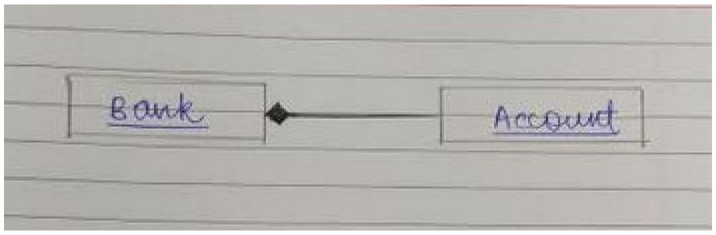


4. **Reflexive Association**
An example here is when a class has many different types of responsibilities. For example, an employee of a company can be an executive, assistant manager, or a CEO. There is no symbol that can be used here, however, the relation will point back to the same class.

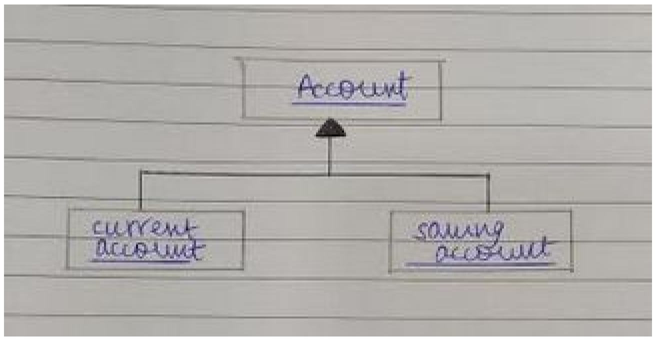
5. **Aggregation**
When a class is formed as a collection of other classes, it is called an aggregation relationship between these classes. It is also called a “has a” relationship.



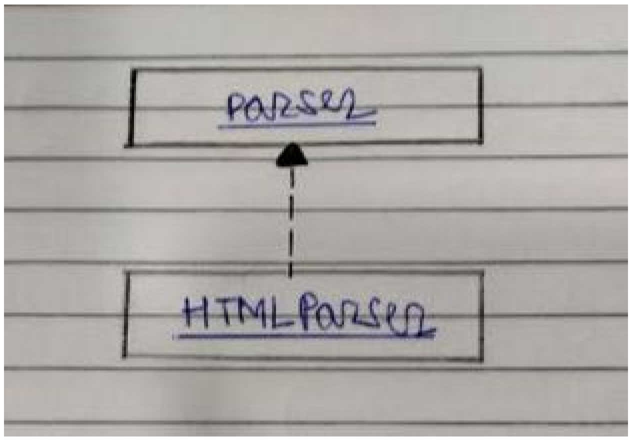
6. **Composition**
The composition is a variation of the aggregation relationship. Composition illustrates that a strong life cycle is present between the classes. Another class diagram relationship that not many are aware of and few really understand.



7. **Generalization/Inheritance**
Known as an “is a” relationship since the child class is a type of the parent class. Generalization is the ideal type of relationship that is used to showcase reusable elements in the class diagram. Literally, the child classes “inherit” the common functionality defined in the parent class.



8. **Realization**
In a realization relationship, one entity (normally an interface) defines a set of functionalities as a contract and the other entity (normally a class) “realizes ” the contract by implementing the functionality defined in the contract.



Q4. Differentiate between include and extend.

Extend is used when a use case conditionally adds steps to another first class use case. For example, imagine "Withdraw Cash" is a use case of an ATM machine. "Assess Fee" would extend Withdraw Cash and describe the conditional "extension point" that is instantiated when the ATM user doesn't bank at the ATM's owning institution. Notice that the basic "Withdraw Cash" use case stands on its own, without the extension.

Include is used to extract use case fragments that are duplicated in multiple use cases. The included use case cannot stand alone and the original use case is not complete without the included one. This should be used sparingly an only in cases where the duplication is significant and exists by design (rather than by coincidence). For example, the flow of events that occurs at the beginning of every ATM use case (when the user puts in their ATM card, enters their PIN, and is shown the main menu) would be a good candidate for an include.

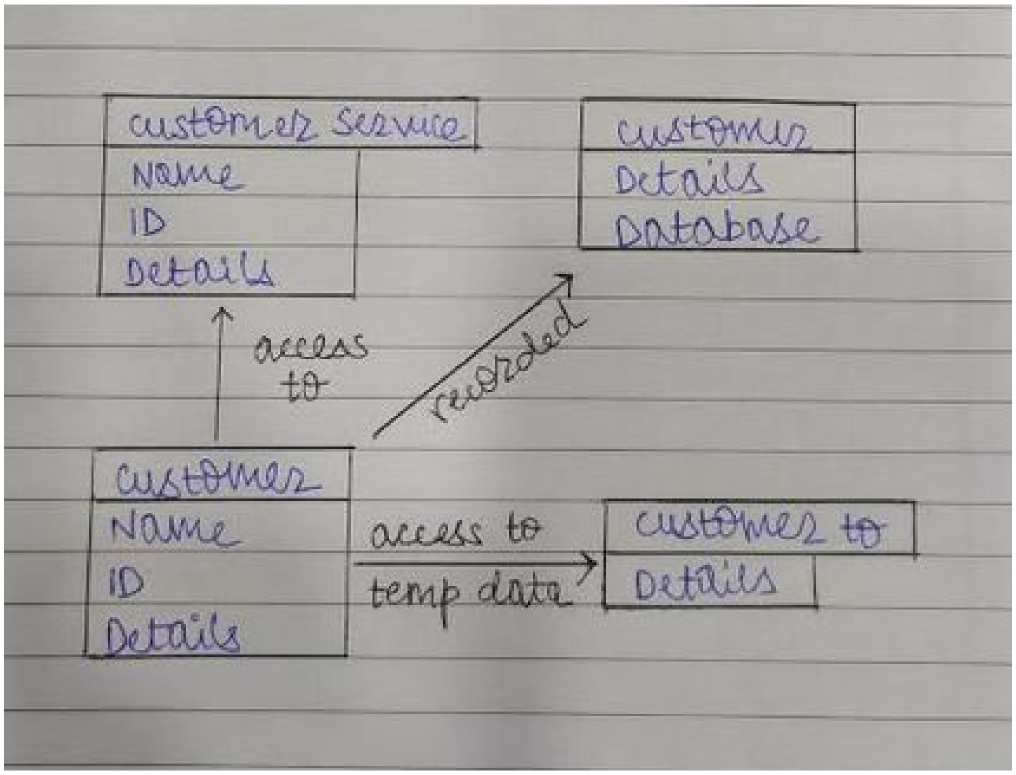
Q5. What is the need of fork and join in activity diagram.

A **fork node** is used to split a single incoming flow into multiple concurrent flows. It is represented as a straight, slightly thicker line in an activity diagram.

A **join node** joins multiple concurrent flows back into a single outgoing flow. A fork and join mode used together are often referred to as synchronization.

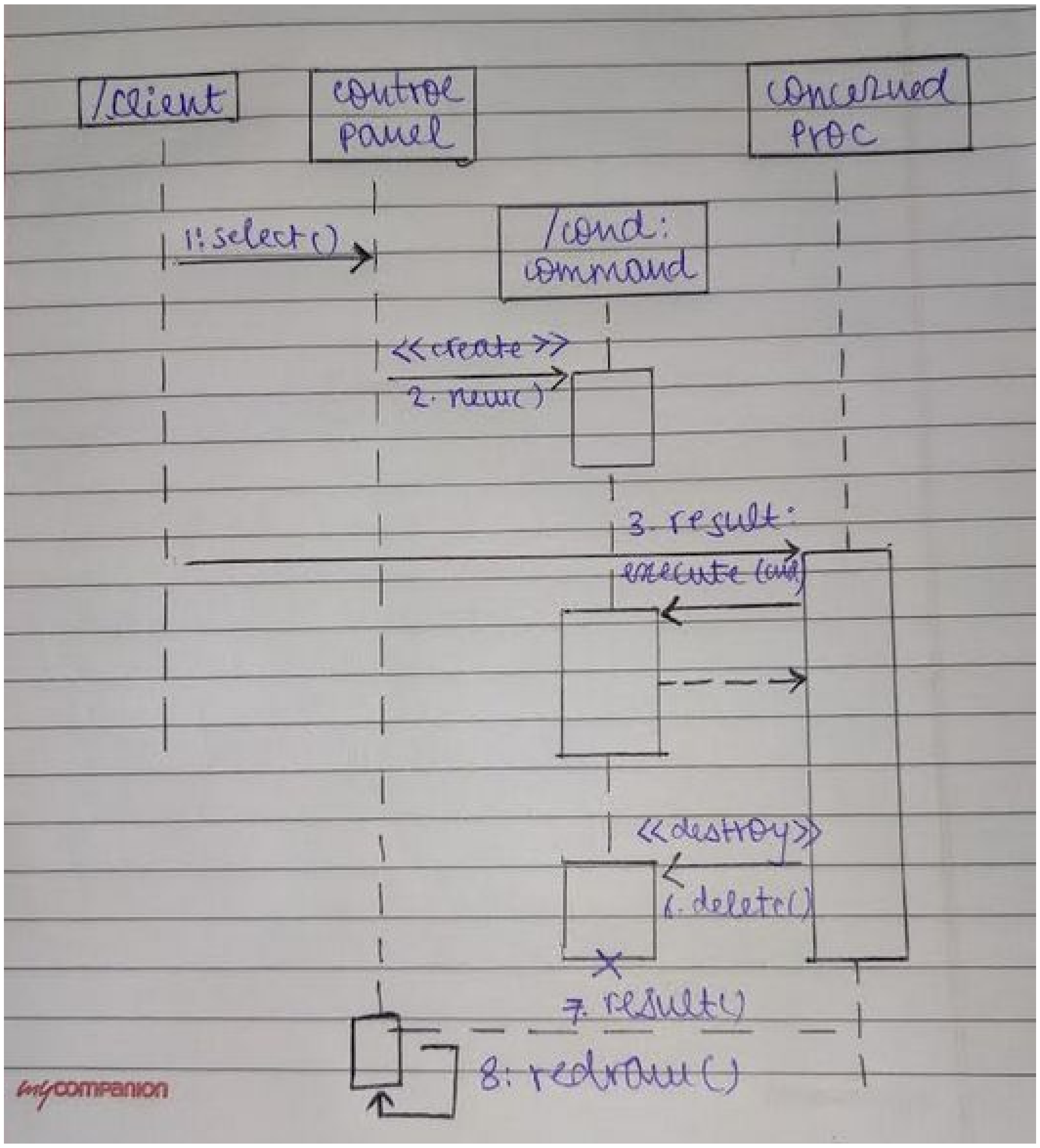
Q6. Draw package diagram for costumer data access having 4 classes customer service,customer , customerDAO and customerTO . Assume the relationships in class to complete the package diagram

The diagrammatic representation is as follows:



Q7. A client selects a menu item on a control panel. In response, the control panel creates a command and asks a command processor to execute it. The command processor asks the command to execute itself, stores the result, deletes the command, then returns the result to the control panel to be displayed. draw its sequence diagram.

The diagrammatic representation is as follows:


/client
control panel
concerned proc
/cmd: command
1: select()
<<create>>
2. new()
3. result: execute (cmd)
<<destroy>>
6. delete()
7. result()
8: redraw()