Jai Swaminarayan

40024

OOAD) cl-3 Ajinkya

PAGE NO.
DATE :

Cohesion

In computer programming cohesion refers to the degree to which the elements inside a module belong together.

→ In one sense, it is a measure of strength of relationship between methods and data of a class and some unifying purpose or concept served by that class.

→ It is a measure of strength of relationship between class & method and data themselves.

Types of Cohesion

1. Coincidental Cohesion

This is worst Cohesion.

It occurs when parts of a module are grouped arbitrarily.

(2.) Logical cohesion

It occurs when parts of module are grouped together & because they are logically categorized to do the same thing even though they are different by nature.

(3.) Temporal Cohesion

It is when parts of module are grouped when they are processed. The parts are processed at particular time in program execution.

(4.) Procedural Cohesion

Procedural cohesion is when parts of module are grouped because they follow a certain sequence of execution.

(5) Communication / Informational cohesion

Communication cohesion is when parts of a module are grouped because they operate on same data.

(6) Sequential Cohesion

when parts of module are grouped because output from one part is input to another part like assembly line

(7) Functional Cohesion (Best)

Modules are grouped because they all contribute to single well-defined task of a module.

01-4

CRC Cards

→ CRC is a abbreviated form of Class Responsibility Collaboration.

→ It is a brainstorming tool used in the design of object oriented software.

→ They were originally proposed by Ward Cunningham and Kent Beck as a teaching tool, but are also popular among expert designers.

→ They are recommended by extreme programming supporters.

→ If you want to explore multiple alternative interactions quickly, you may be better off with CRC cards, as that avoids a lot of drawing and erasing.

→ It is often handy to have a CRC card session to explore

design alternatives and then use sequence diagram to capture any interactions that you want to refer later.

- CRC cards are created for index cards. Members of brainstorming session will write up one CRC card for each relevant class/object of their design.
- The card is partitioned in 3 areas.

1. On top of card, class name
2. On the left, responsibility of class
3. On the right, collaborators with which class interacts to fulfill its responsibilities.

Q1-6

PAGE NO.
DATE :

Dialogue Metaphor

→ The direct manipulation and document methaphors emphasize objects in computer with which user interacts.

→ Another view of interface is "dialog methaphor" in which interacting with computer is much like carrying a conversation or dialog.

→ In fact, user interface design is often referred to as "dialog design".

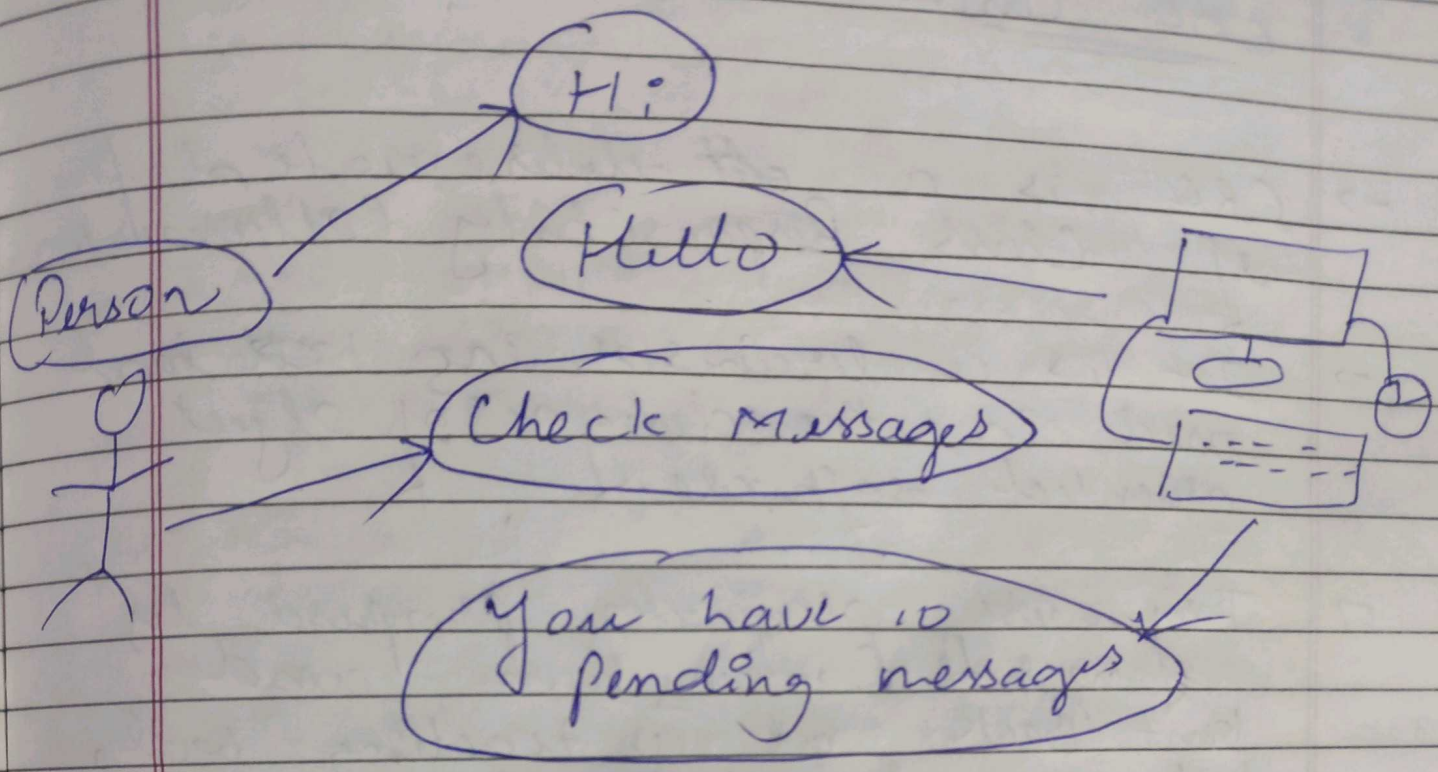
→ Carrying on a dialog, or conversation requires each participant to listen to and respond to questions and comments for the other, exchanging information in a sequence.

PAGE NO.
DATE :

→ The dialog metaphor is another way of thinking about Human-computer interaction because computer "listens to" and responds to user questions or comments, and user "listens to" and listens to and respond to user question or comments.

→ Like the direct manipulation metaphor, the dialog metaphor is based on object-oriented view of system because communication involves message from one object to another.

→ Next figure shows how user and computer can communicate by sending message to each other


Hi
Hello
Person
Check Messages
You have 10 pending messages

Q1 ->

PAGE NO.
DATE :

Significance of Deployment Diagram

-> The term deployment itself describes the purpose of diagram.

-> Deployment diagrams are used for describing the hardware components, where software components are deployed.

-> Component diagrams and deployment diagrams are closely related.

-> Component diagrams are used to describe the components and deployment shows how they are deployed in hardware.

-> UML is mainly designed to focus on software artifacts of system. However these 2 diagrams are special diagrams used to focus on software and hardware components.

Teacher's Signature.....

PAGE NO.
DATE :

→ Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of system.

→ They are used by system engineers.

→ It is also mainly used for following 3 reasons:-

→ They are used to visualize hardware topology of system.

→ They are used to describe the hardware components used to deploy software components.

→ It is also helpful to describe the runtime processing nodes.

Teacher's Signature.....

Jai Swaminarayan

40024 - Ajinkya Rathod

PAGE NO.
DATE :

OOAD.

Q2 - (1)

1. Relational Database vs Object Oriented

RDBMS	OO DBMS
Stands for relational database management system.	Stands for Object Oriented Database Management System.
Stores data in Entities defined as tables hold specific information.	Stores data as objects.
Handles comparatively simpler data	Handles large and complex data than RDBMS.
Entity type refers to the collection of entity that share a common definition.	Class describes a group of objects that share common relationship behaviours and also similar properties

PAGE NO.
DATE :

RDBMS	OODBMS
RDBMS stores only data.	stores data as well as methods to use it.
Data independent from application program	Data Encapsulation.
A primary key distinctively identifies an object in a table.	An object identifier is unambiguous, long-term name for any type of object or entity.
The data and relationships are represented by collection of inter-related tables.	The data is presented in form of ~~tables~~ objects.
It enables the user to create, update	It enables to store complex data.

PAGE NO.
DATE :

RDBMS	OODBMS
administer and interact with relational database.	
RDMS is basis for SQL and for all modern database system such as -> MS My SQL -> IBM -> DB2 -> Oracle -> My SQL -> Microsoft Access	OODBMS is examples of -> Versant Object Database -> Objectivity DB -> Object store -> Cache -> ZODB

Q2 - (2)

PAGE NO.
DATE :

There are 5 types of design class each each to the aspects of the layer of design architecture

1. Uses interface classes.

→ These classes are designed for Human Computer Interaction

→ These interface classes define all abstraction which is required for HCI.

2. Business Domain Classes

These class are commonly refiners of analysis classes.

They are recognized as attributes and methods which are required to implement the elements of business domain.

3. Process class

It implements the lower level business abstraction which is needed to completely manage the business domain class.

4. Persistance Class

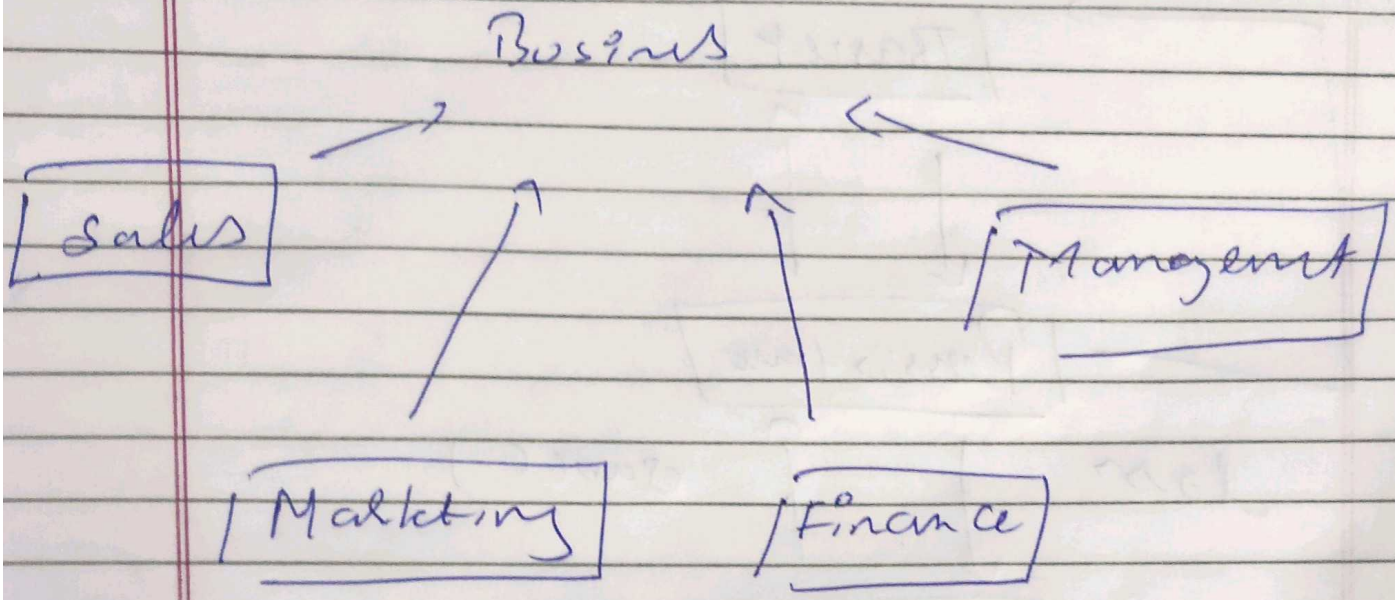
It shows data stores that will persist behind execution of software.

i. System Classes

Any 4 - do not explained

PAGE NO.
DATE :

Business Domain



Process Class

- «interface»
  Owns
  + acquist
  + dispose

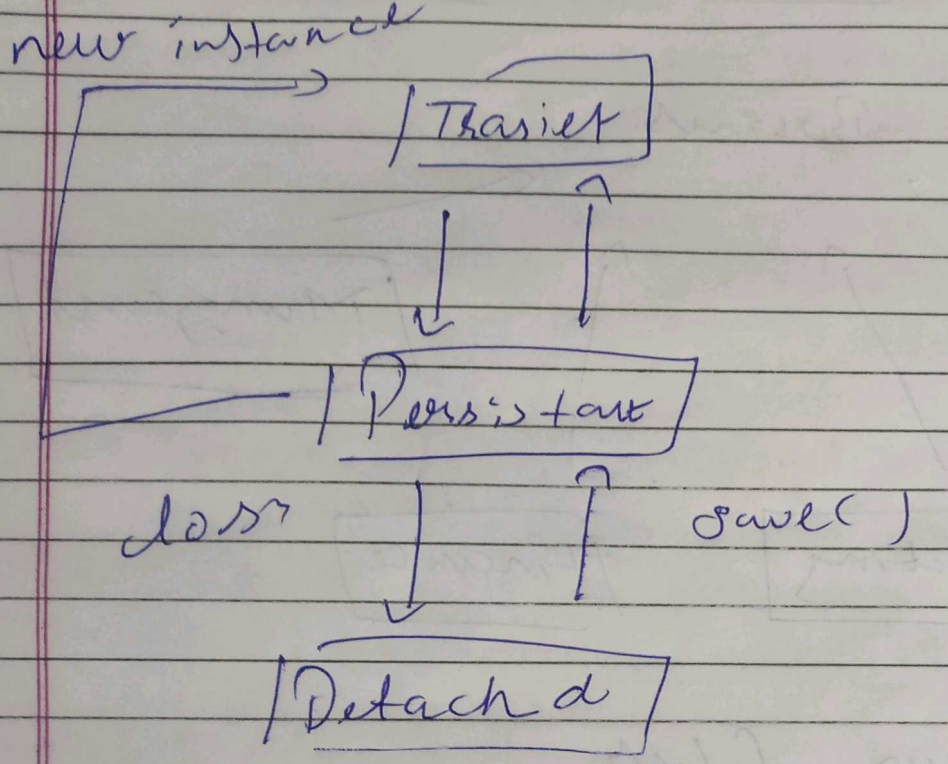
class | Person
- real
- tangible

Corporation
- current
- fixed

PAGE NO.
DATE :

Persistant Class

~~Persistant Class~~



User Interface Class