


# Functional Modeling

- 
- ▶ Functional Modelling gives the process perspective of the object-oriented analysis model
  - ▶ An overview of what the system is supposed to do.
  - ▶ It defines the function of the internal processes in the system with the aid of Data Flow Diagrams (DFDs).
  - ▶ It depicts the functional derivation of the data values without indicating how they are derived when they are computed, or why they need to be computed.

# Data Flow Diagrams

- ▶ A data flow diagram is a graph which shows the flow of data values from their sources in objects through processes that transform them to their destinations on other objects.

The four main parts of a DFD are –

- ▶ Processes,
- ▶ Data Flows,
- ▶ Actors, and
- ▶ Data Stores.

The other parts of a DFD are –

- ▶ Constraints, and
- ▶ Control Flows.

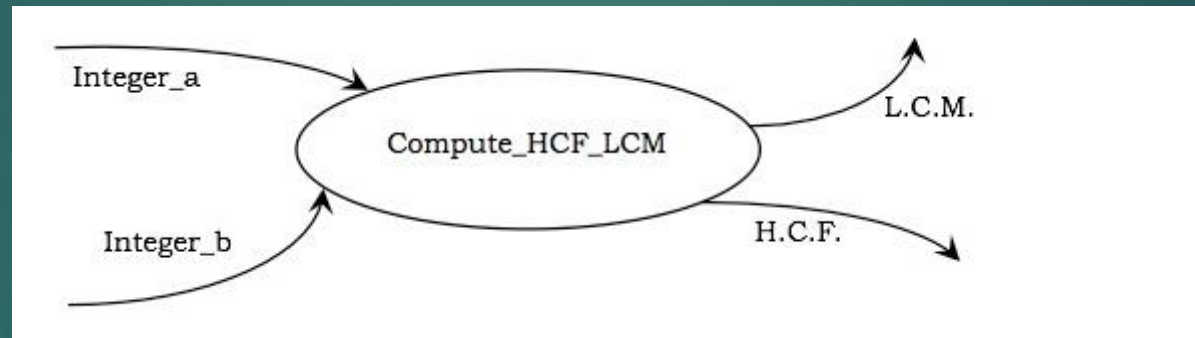
# Processes

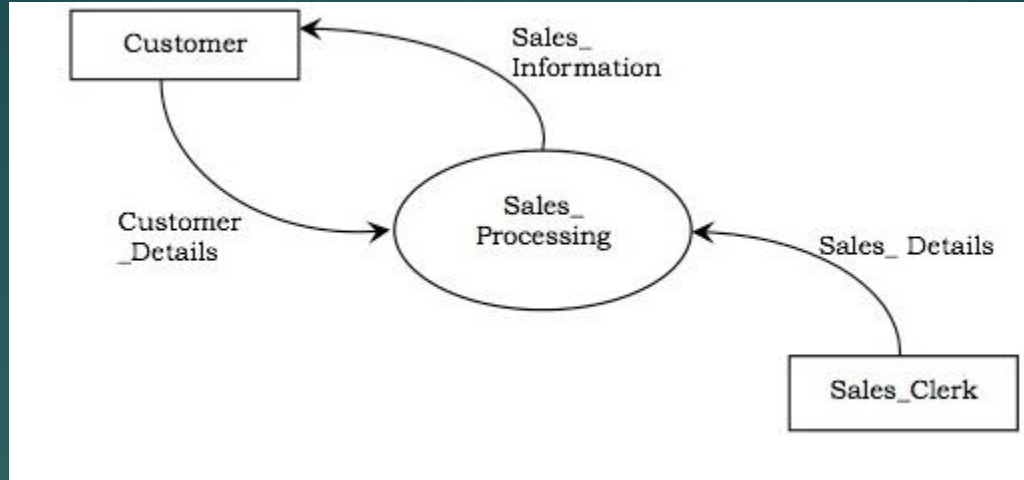
- ▶ Processes are the computational activities that transform data values. A whole system can be visualized as a high-level process.
- ▶ A process may be further divided into smaller components. The lowest-level process may be a simple function.

# Representation in DFD

- ▶ A process is represented as an ellipse with its name written inside it and contains a fixed number of input and output data values.

- ▶ Data Flows
- ▶ Data flow represents the flow of data between two processes. It could be between an actor and a process, or between a data store and a process. A data flow denotes the value of a data item at some point of the computation. This value is not changed by the data flow.
- ▶ **Representation in DFD** – A data flow is represented by a directed arc or an arrow, labelled with the name of the data item that it carries.
- ▶ In the above figure, Integer\_a and Integer\_b represent the input data flows to the process, while L.C.M. and H.C.F. are the output data flows.
- ▶ A data flow may be forked in the following cases –
- ▶ The output value is sent to several places as shown in the following figure. Here, the output arrows are unlabelled as they denote the same value.
- ▶ The data flow contains an aggregate value, and each of the components is sent to different places as shown in the following figure. Here, each of the forked components is labelled






- ▶ Actors
- ▶ Actors are the active objects that interact with the system by either producing data and inputting them to the system, or consuming data produced by the system. In other words, actors serve as the sources and the sinks of data.
- ▶ **Representation in DFD** – An actor is represented by a rectangle. Actors are connected to the inputs and outputs and lie on the boundary of the DFD.
- ▶ **Example** – The following figure shows the actors, namely, Customer and Sales\_Clerk in a counter sales system.





- ▶ Data Stores
- ▶ Data stores are the passive objects that act as a repository of data. Unlike actors, they cannot perform any operations. They are used to store data and retrieve the stored data. They represent a data structure, a disk file, or a table in a database.
- ▶ **Representation in DFD** – A data store is represented by two parallel lines containing the name of the data store. Each data store is connected to at least one process. Input arrows contain information to modify the contents of the data store, while output arrows contain information retrieved from the data store. When a part of the information is to be retrieved, the output arrow is labelled. An unlabelled arrow denotes full data retrieval. A two-way arrow implies both retrieval and update.
- ▶ **Example** – The following figure shows a data store, Sales\_Record, that stores the details of all sales. Input to the data store comprises of details of sales such as item, billing amount, date, etc. To find the average sales, the process retrieves the sales records and computes the average.

- 
- ▶ Constraints
  - ▶ Constraints specify the conditions or restrictions that need to be satisfied over time. They allow adding new rules or modifying existing ones. Constraints can appear in all the three models of object-oriented analysis.
  - ▶ In Object Modelling, the constraints define the relationship between objects. They may also define the relationship between the different values that an object may take at different times.
  - ▶ In Dynamic Modelling, the constraints define the relationship between the states and events of different objects.
  - ▶ In Functional Modelling, the constraints define the restrictions on the transformations and computations.
  - ▶ **Representation** – A constraint is rendered as a string within braces.