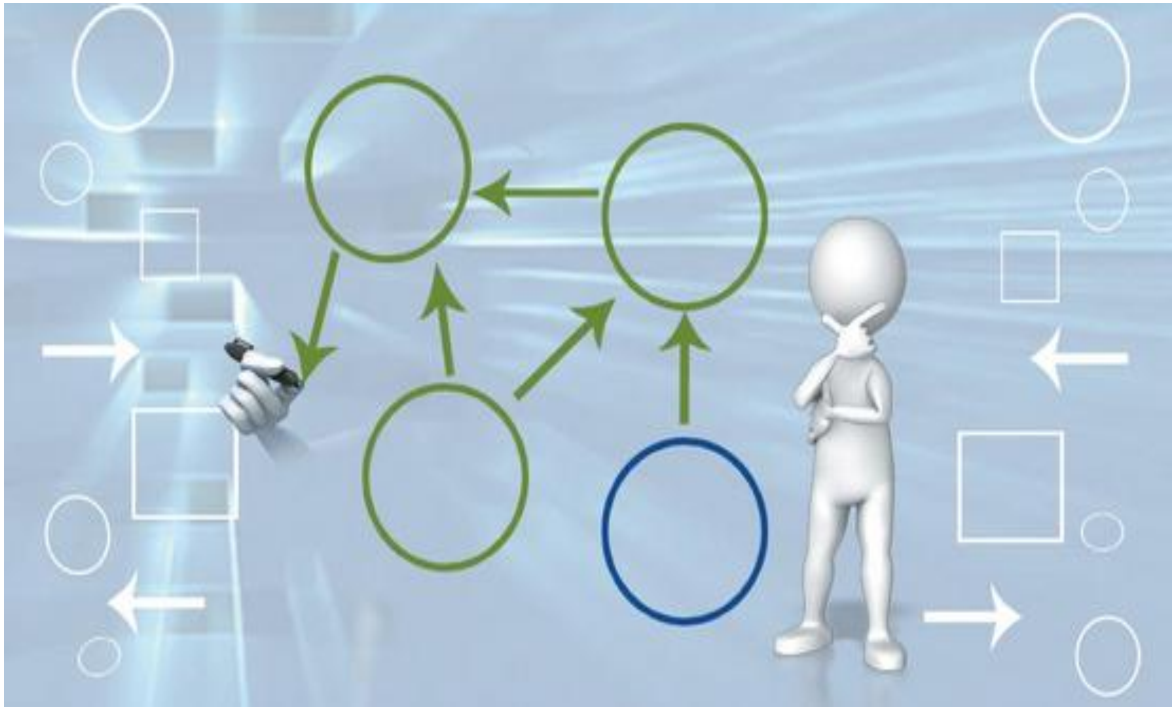
Topic – Data Flow Diagram(DFD)



Presented By –

M.M. Rakibul Hasan

Faculty, CSE, IUBAT University

Introduction to DFD

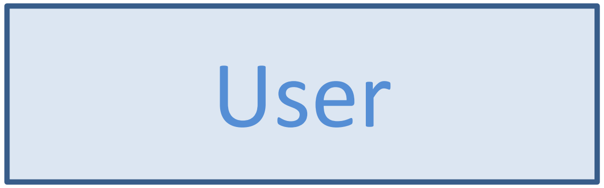
- What is DFD?

  - A data flow diagram is a graphical depiction of flow of data through intended software system and is used as 1st step to create an overview of system.

  - It's really useful as it provides overview of data as well as functionality to software designers

Components Of DFD: **Entity**

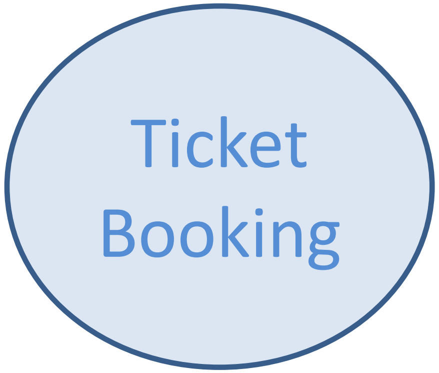
- External Entities:
  - They could be a person (facebook users), another software(like facebook) or a hardware (sensors) which provide to or consume information from the intended software.
  - Represented by rectangle:
    - Must be named
    - No direct data flow between two entities ever.

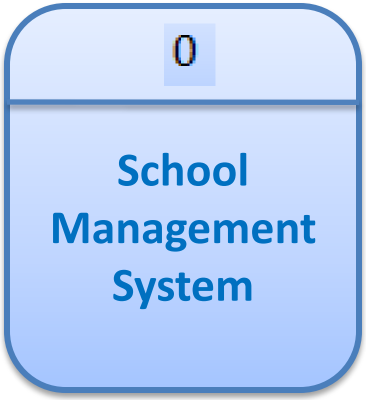


User

Components Of DFD: **Process**

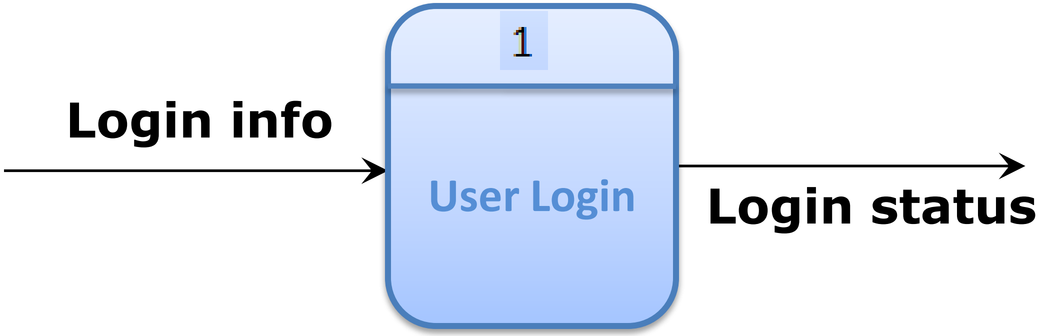
- A circle (sometimes called a *bubble) represents a process or transform that* is applied to data and changes it in some way.
- The basic rules:
  - It must be properly labeled
  - It must not be repeated in a diagram





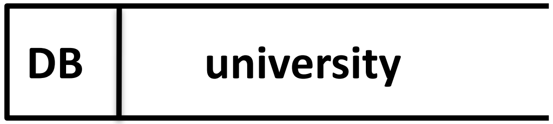
Components Of DFD: **Data Flow**

- The basic rules:
  - Data flows can’t be bidirectional, i.e the input data flow and the output data flow for a process, data store or for an entity should always be different.
  - The data flows should always be labelled
  - The labels should be precise and informative
  - You can join two similar input data flows(join) or two similar output data flows (fork)



Components Of DFD: **Data Store**

- Data stores are places where data may be stored. This information may be stored either temporarily or permanently by the user.
- They are internal to the system.
- The basic rules:
  - Never shown in context level diagram
  - No direct data flows between two data sources
  - Symbol:



DFD General rules

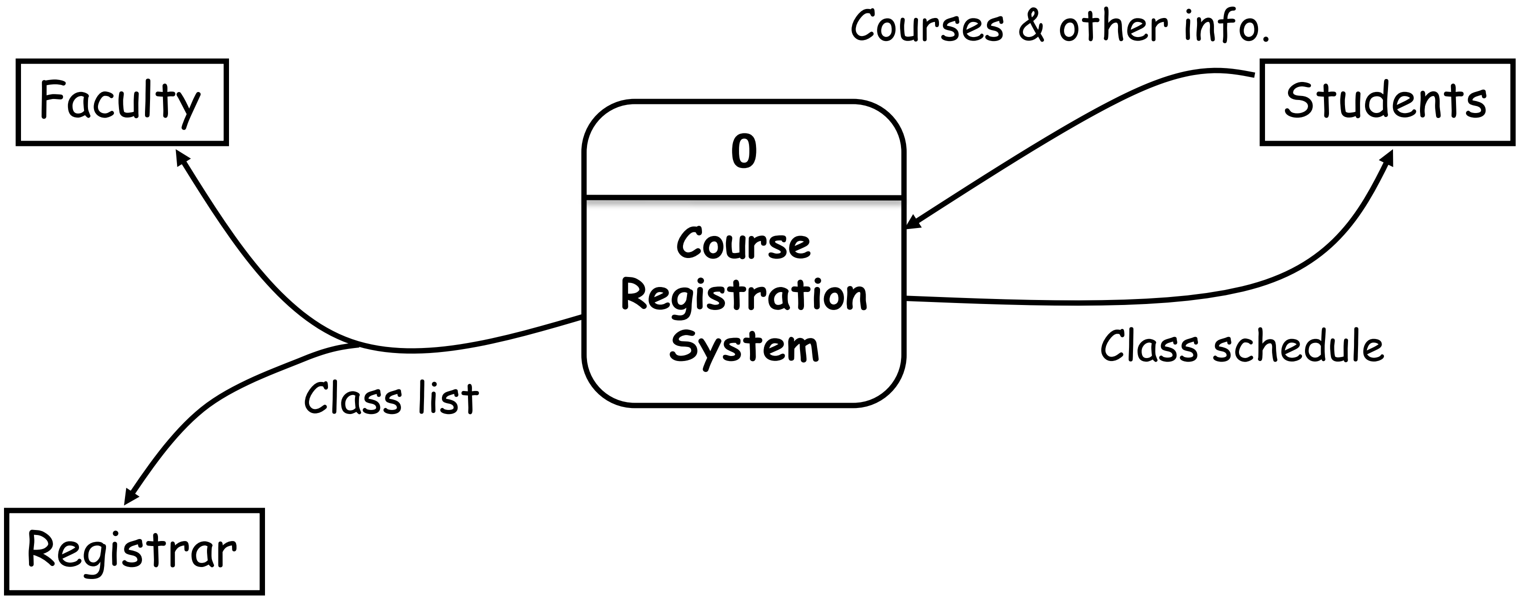
- Basic rules that apply to all DFDs:
  - No internal logic should be shown like loops, if-else, this is not a flow chart
  - In order to keep the diagram uncluttered, you can repeat data stores and external entities
  - No process can have only output data flows (a miracle).
  - No process can have only input data flows (black hole).
  - Data cannot be moved directly from one store to another without a process.
  - Data cannot move directly from an external entity to a data store without a process.
  - Data stores can’t be sink(only input data flows) or source (only output data flows) in level 1 DFD

Context Level Diagram

- A level 0 DFD, also called a fundamental system model or a context model.
- It represents the entire software element as a single process with input and output data.
- All the external entities should be identified and shown.
- Rule:
  - Only one process
  - Data flows should be labeled.
  - No data store can be shown in context diagram

Context Level Diagram for CRS

Context Diagram for Course Registration System:

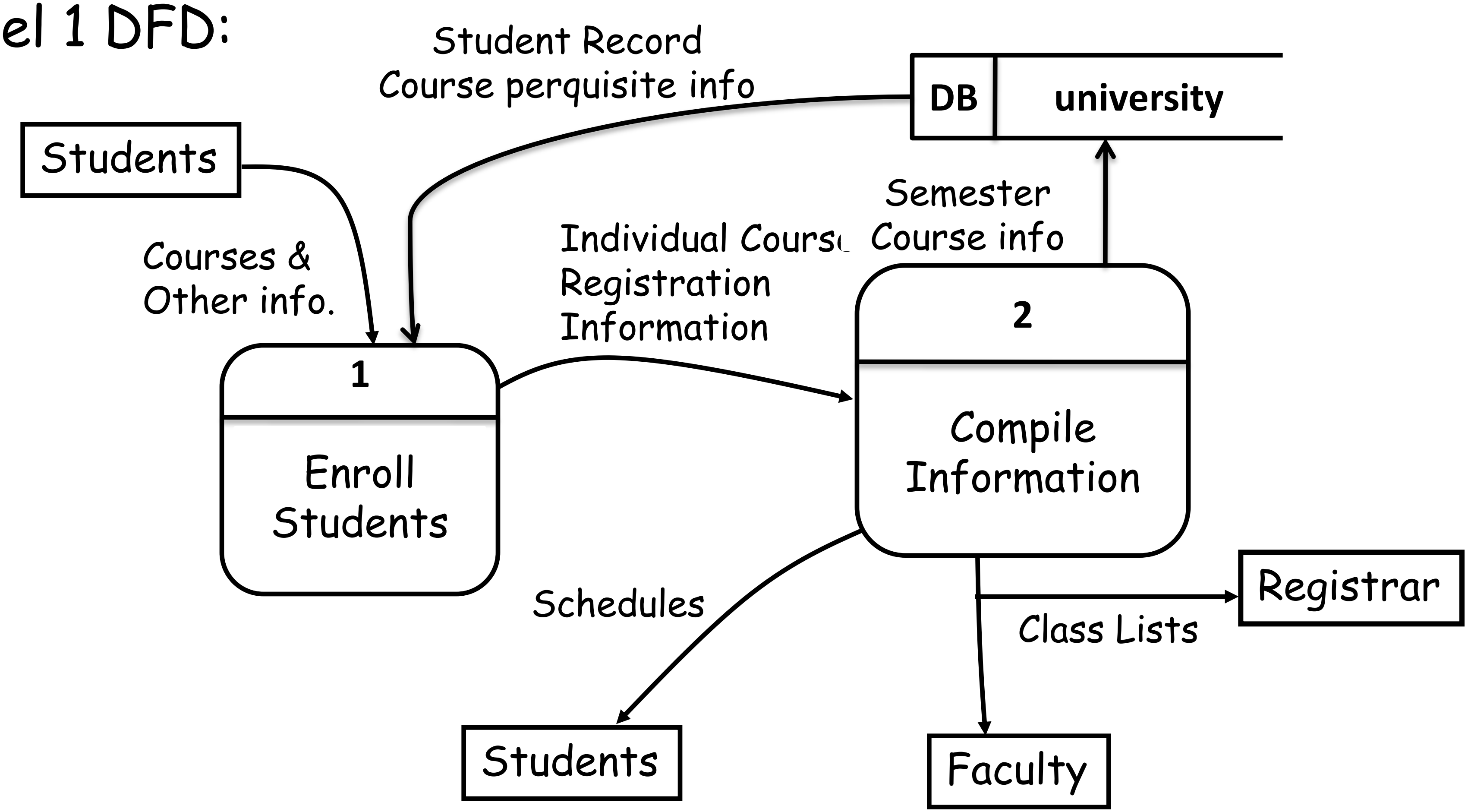


Level 1 DFD

- The level 1 DFD we construct is a more refined version of the context diagram.
- It covers the entire system, all the main processes are shown
- The DFD should be balanced with respect to context diagram
  - No new external entities should be there
  - The data flows from context diagrams should be visible
- Rules:
  - It should consists of 5-9 processes(bubbles).
  - Repetition of data sources is allowed.
  - Process can not be repeated.

Level 1 DFD for CRS

Level 1 DFD:



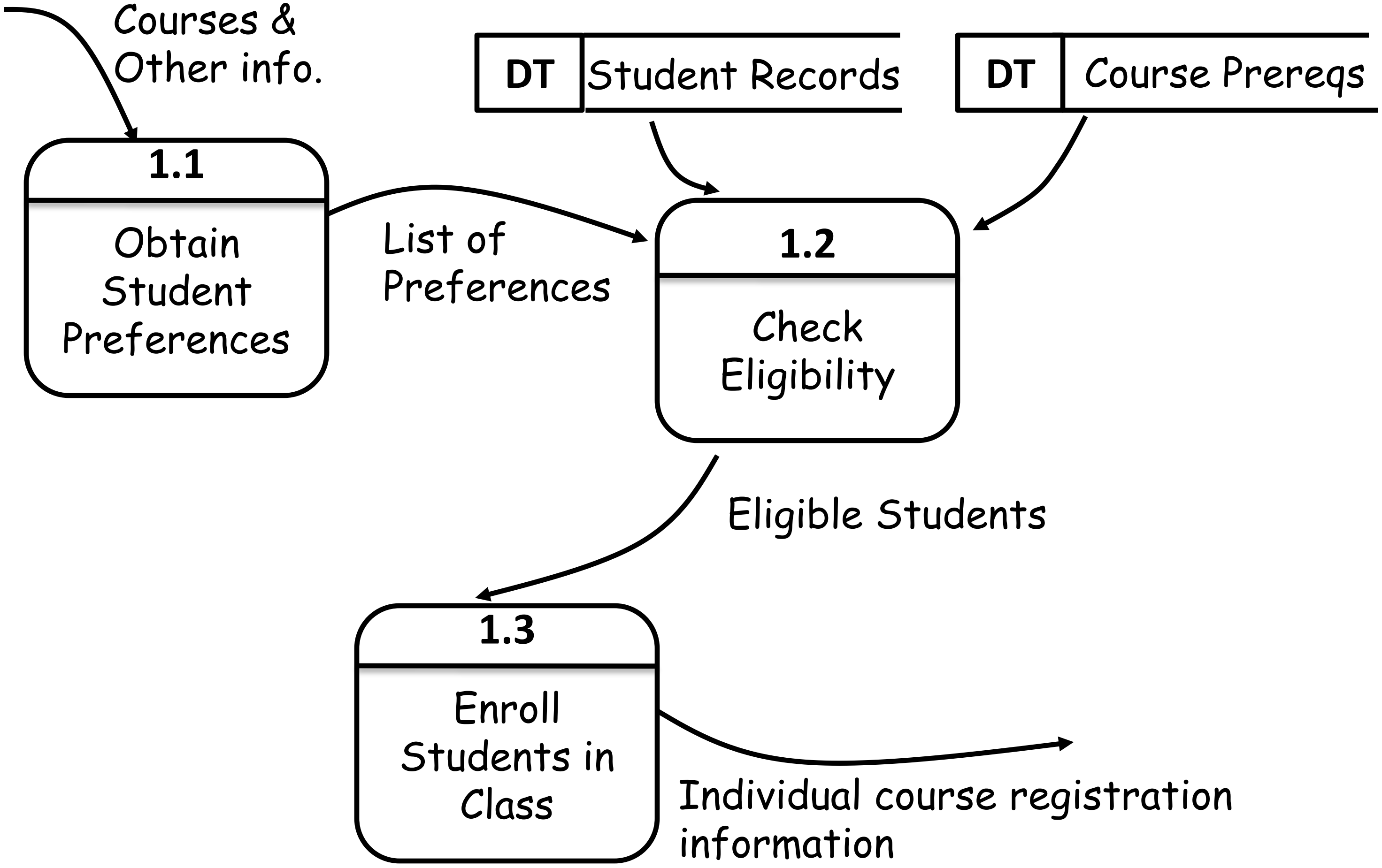
Note: External entity Students is replicated to avoid crossing lines

Level 2 DFD

- Only those processes that merit being expanded need to have level 2 DFDs .
- Level 2 DFD completely describes any one process from the level 1 DFD.
- Rules:
  - All the data flows into and out of selected process on the level 1 DFD also appear on the level 2 DFD
  - Repetition of data sources is allowed.
  - A Data store can appear as a sink or source within level 2 DFD

Level 2 DFD for CRS

Level 2 DFD Process 1 (Enroll Students):



Check List

- There are many errors that may occur when drawing data flow diagrams.
  - External entities must be people or systems that send information to or accept information form the system to be engineered
  - Check the direction of data flows to and from data stores
  - Data flows must always be labelled with the data they contain. Do not put verbs in the data flow description as this implies a process

Check List Continued..

- Parent and child diagrams should be consistent.
- Make sure each process has at least one input and one output.
- Each data store should have at least one input and one output on the DFDs somewhere.
- Each process name should start with a verb

Thank You