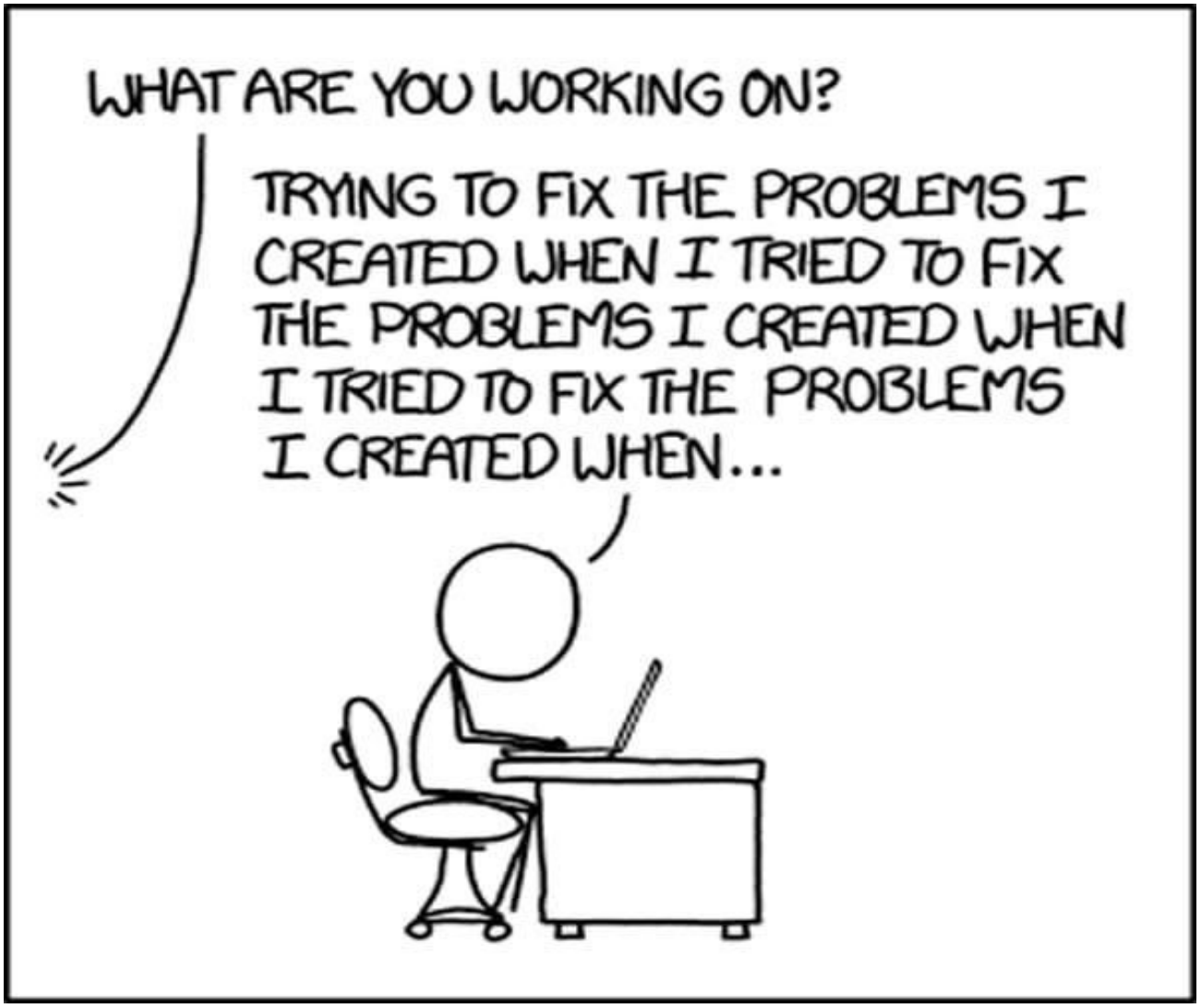
Lecture V:

System Modeling

General Views



Topics Covered

- Systems Modeling
- Data and Process Modeling
- Data Flow Diagrams
- Context Diagrams

System Modeling

System Modeling

- This is the process of **developing abstract models** of a system, with **each model** presenting a **different view** or **perspective** of that system.

- System modeling has now come to mean **representing a system using some kind of graphical notation**, which is now almost always based on notations in the **Unified Modeling Language (UML)**.

- System modeling **helps the analyst**
  - to **understand** the **functionality** of the system and
  - models are used to **communicate with customers**.

The Unified Modeling Language (UML)

- Systems analysts **use UML** to describe **O-O systems**
- UML uses a **set of symbols** to represent **graphically** the various **components** and **relationships** within a **system**
- Although the UML can be used for business process modeling and requirements modeling, it is **mainly used to support O-O system** analysis and to **develop object models**

Existing and Planned System Models

- Models of the existing system are **used during requirements engineering**.
  - They help **clarify** what the existing system does and can be
  - Used as a **basis for discussing** its strengths and weaknesses.
  - These then **lead to requirements** for the new system.

- Models of the new system are used during requirements engineering to help **explain the proposed requirements to other system stakeholders**.
  - Engineers use these models to discuss design proposals and
  - to document the system for implementation.

- In a model-driven engineering process, it is **possible to generate a complete or partial system implementation from the system model**.

System Perspective

- An **external perspective**, where you model the **context or environment** of the system. (For example DFDs, Context Diagrams)
- An **interaction perspective**, where you model the **interactions** between a **system** and its **environment**, or between the **components** of a **system**. (For example ERDs)
- A **structural perspective**, where you model the **organization** of a system or the **structure of the data** that is processed by the system. (For example Object Relationship Diagrams)
- A **behavioral perspective**, where you model the **dynamic behavior** of the system and how it responds to events. (For example Use cases, Sequence Diagram, Activity Diagrams etc)

Data Flow Diagrams (DFDs)

Data Flow Diagram

- A data flow diagram (DFD) **uses various symbols** to show how the system **transforms** input **data** into useful **information**

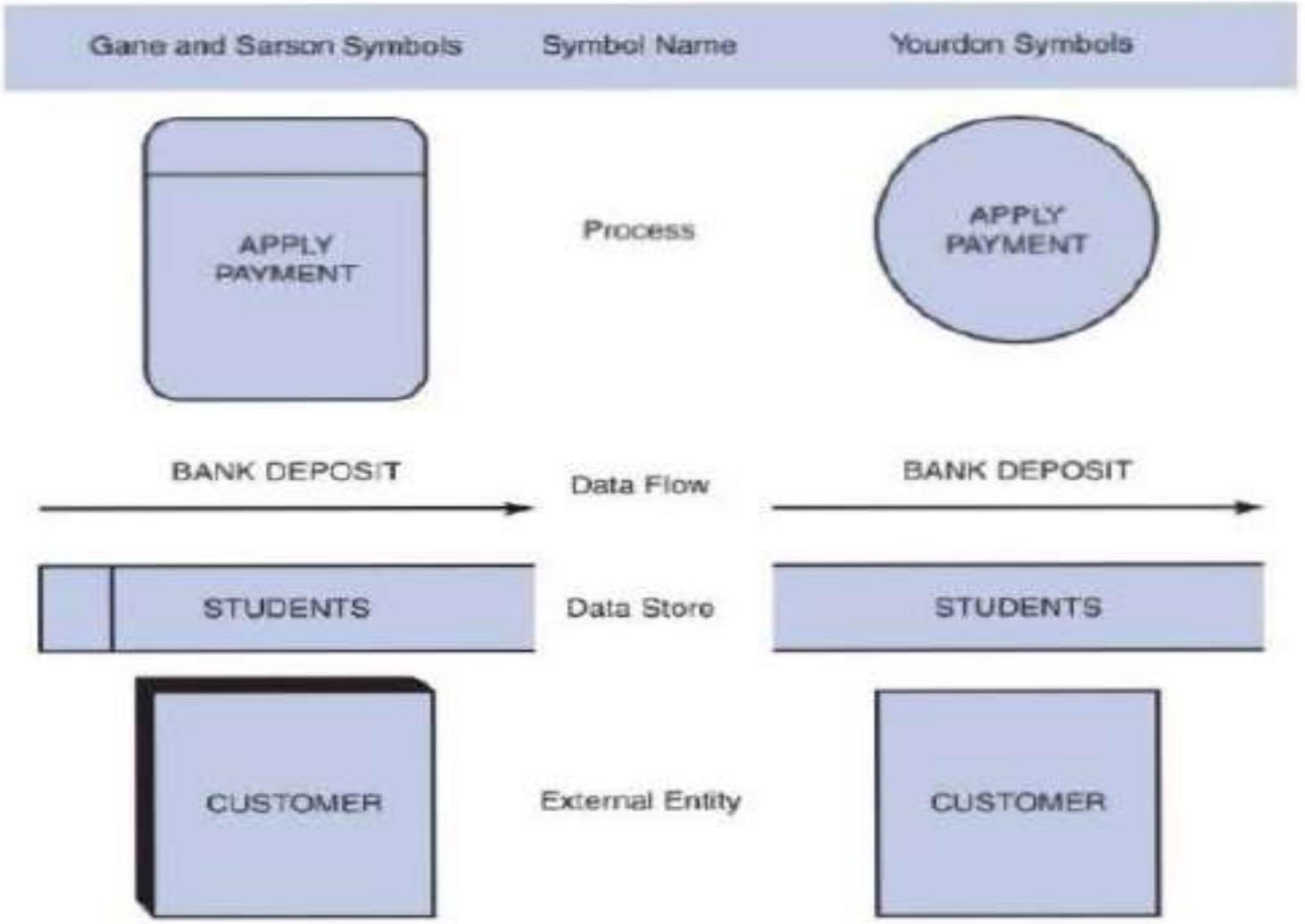
- A DFD **shows** how **data moves** through an information system but **does not show** program **logic** or **processing** steps.

- A set of DFDs provide a logical model that **shows *what* the system does**, *not how* it does it

Data Flow Diagram

- DFDs use **four basic symbols** that represent
  - process(es), data flows, data stores, and entities
- Several different versions of DFD symbols exist
- In this classes, we shall use the ***Gane*** *and* ***Sarson*** *symbol set*
- Another **popular symbol set** is the ***Yourdon*** *symbol set*
- **Symbols** will be referenced using all **CAPITAL LETTERS** for the **symbol name**

Data Flow Diagram Symbols



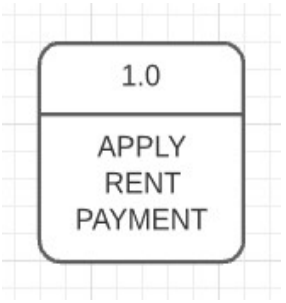
Process Symbols

- A process **receives input data** and **produces output** that has a different content, form, or both

- Processes contain the **business logic/business rules**, that **transforms** the **data** and produces the **required results**

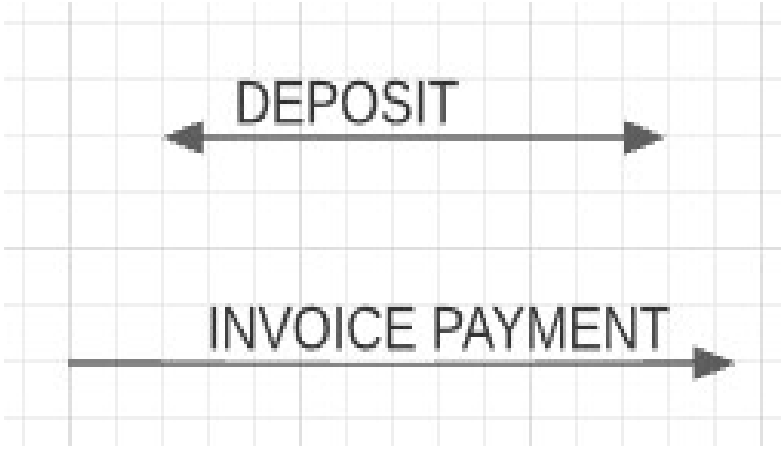
Process Symbols (2)

- The symbol for a process is **a rectangle with rounded corners**
- The **name** of the process appears **inside the rectangle**
- The process name **identifies** a **specific function** and consists of a **verb** (and an adjective, if necessary) followed by a **singular noun**
- Examples of process names are
  - APPLY RENT PAYMENT, CALCULATE COMMISSION, ASSIGN FINAL GRADE, VERIFY ORDER, and FILL ORDER



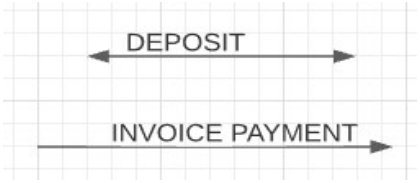
Data Flow Symbol

- A data flow is a **path for data** to move from **one part** of the information system to **another**
- A data flow in a DFD **represents one or more data items**

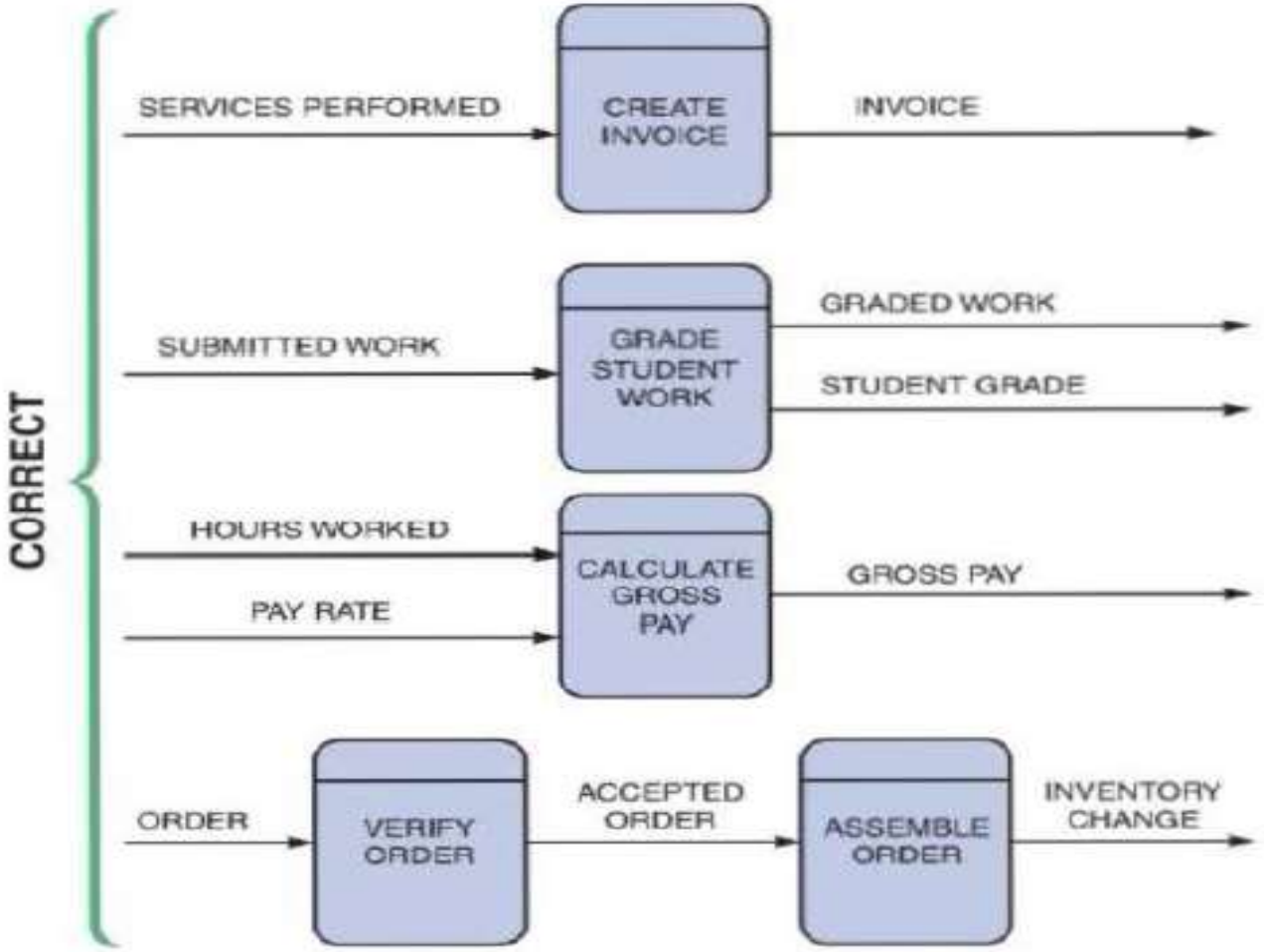


Data Flow Symbol (2)

- The symbol for a data flow is **a line with a single or double arrowhead**
- The data flow **name** appears **above**, **below**, or **alongside** the line
- A data flow name consists of a **singular noun** and/or an **adjective**
- Examples of data flow names are
  - DEPOSIT, INVOICE PAYMENT, STUDENT GRADE, ORDER, and COMMISSION



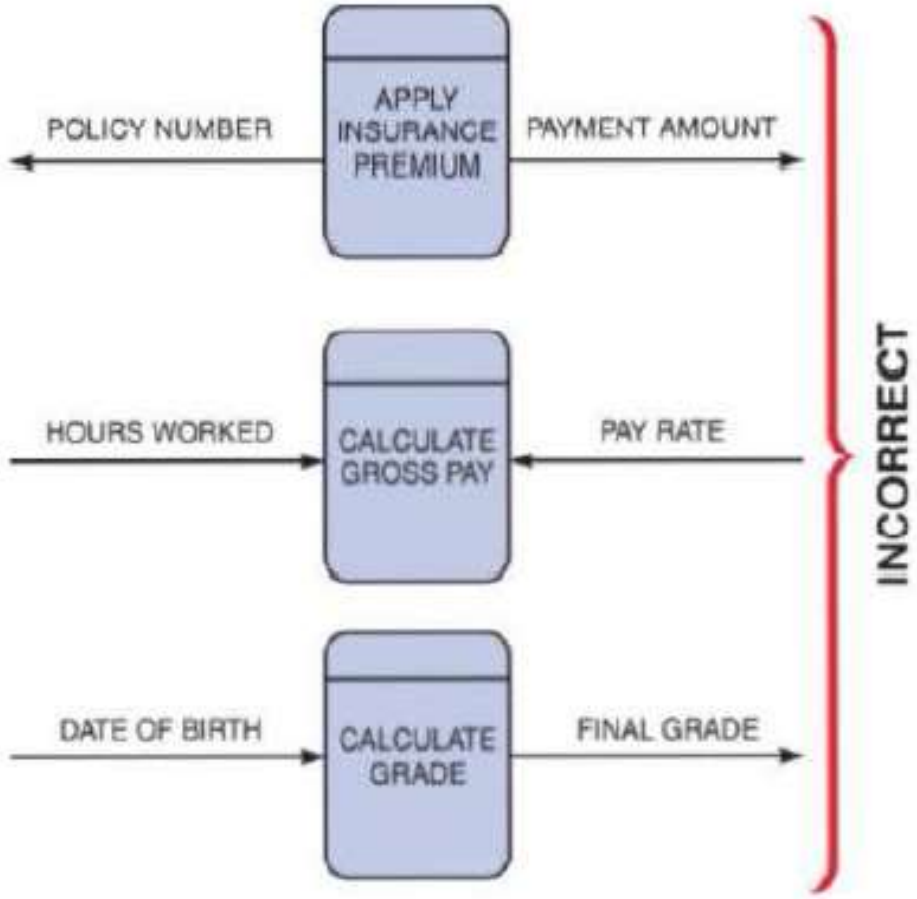
Correct Example Using Data Flow and Process Symbols



INCORRECT Example Using Data Flow and Process Symbols

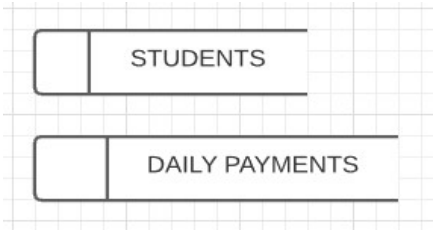
Avoid:

- Spontaneous generation
- Black hole
- Gray hole



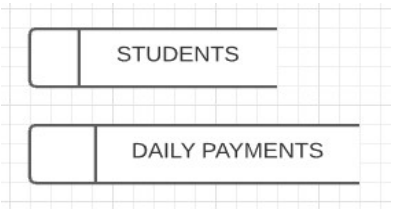
Data Store Symbols

- A data store is used in a DFD to represent **data that the system stores** because **one** or **more processes** need to use the data at a later time
- A DFD **does not show** the **detailed contents** of a data store
- The **physical characteristics** and **length of time** that the data is stored in a data store **are unimportant**



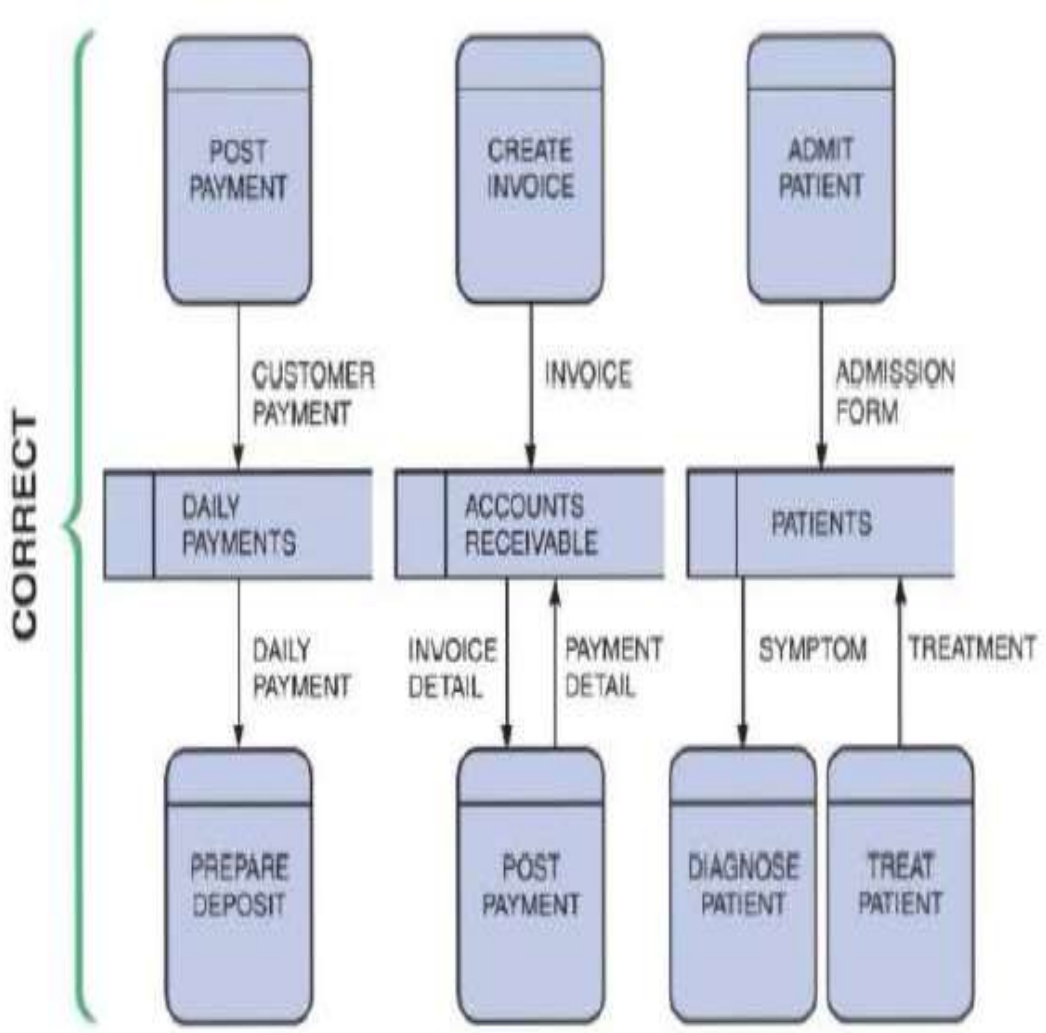
Data Store Symbols (2)

- In a DFD, the **symbol** for a data store is **a flat rectangle that is open on the right side** and **closed** on the **left side**
- The **name** of the data store appears **between** the **lines** and **identifies** the **data** it contains
- A data store name is a **plural name** consisting of a **noun** and/or **adjectives**
- Examples of data store names are
  - STUDENTS, ACCOUNTS RECEIVABLE, PRODUCTS, DAILY PAYMENTS, PURCHASE ORDERS



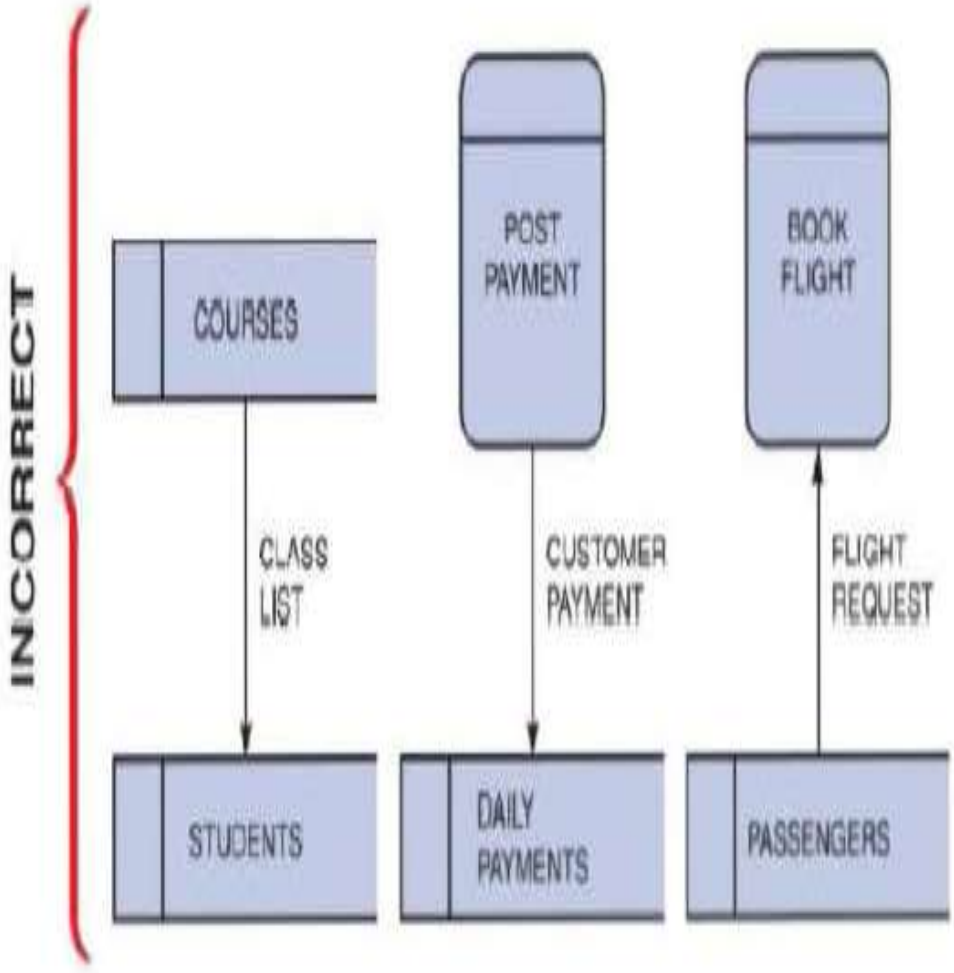
Data Store Symbols (3)

- A data store **MUST be connected to a process** with a **data flow**
- The data store has at **least one incoming** and **one outgoing data flow** and is **connected** to a **process symbol** with a **data flow**



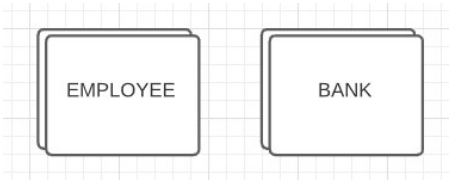
Data Store Symbols (4)

- Violations of the rule (at least one incoming and one outgoing data flow)
  - **Two data stores are connected incorrectly** because no process is between them
  - Also, COURSES has **no incoming data flow** and STUDENTS has **no outgoing data flow**
  - In the second and third examples, the **data stores lack** either an **outgoing** or **incoming data flow**



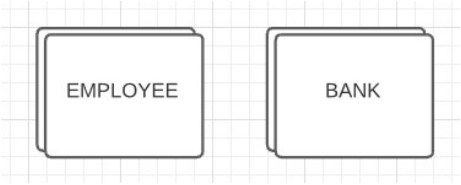
Entity Symbols

- The **symbol** for an entity is a **rectangle**, which **may be shaded** to make it look **three-dimensional**
- The **name** of the entity **appears inside** the symbol
- A DFD shows only **external entities** that **provide data** to the system or **receive output** from the system
- An external entity **shows** the **boundaries** of the system and how the **system interfaces** with the **outside world**

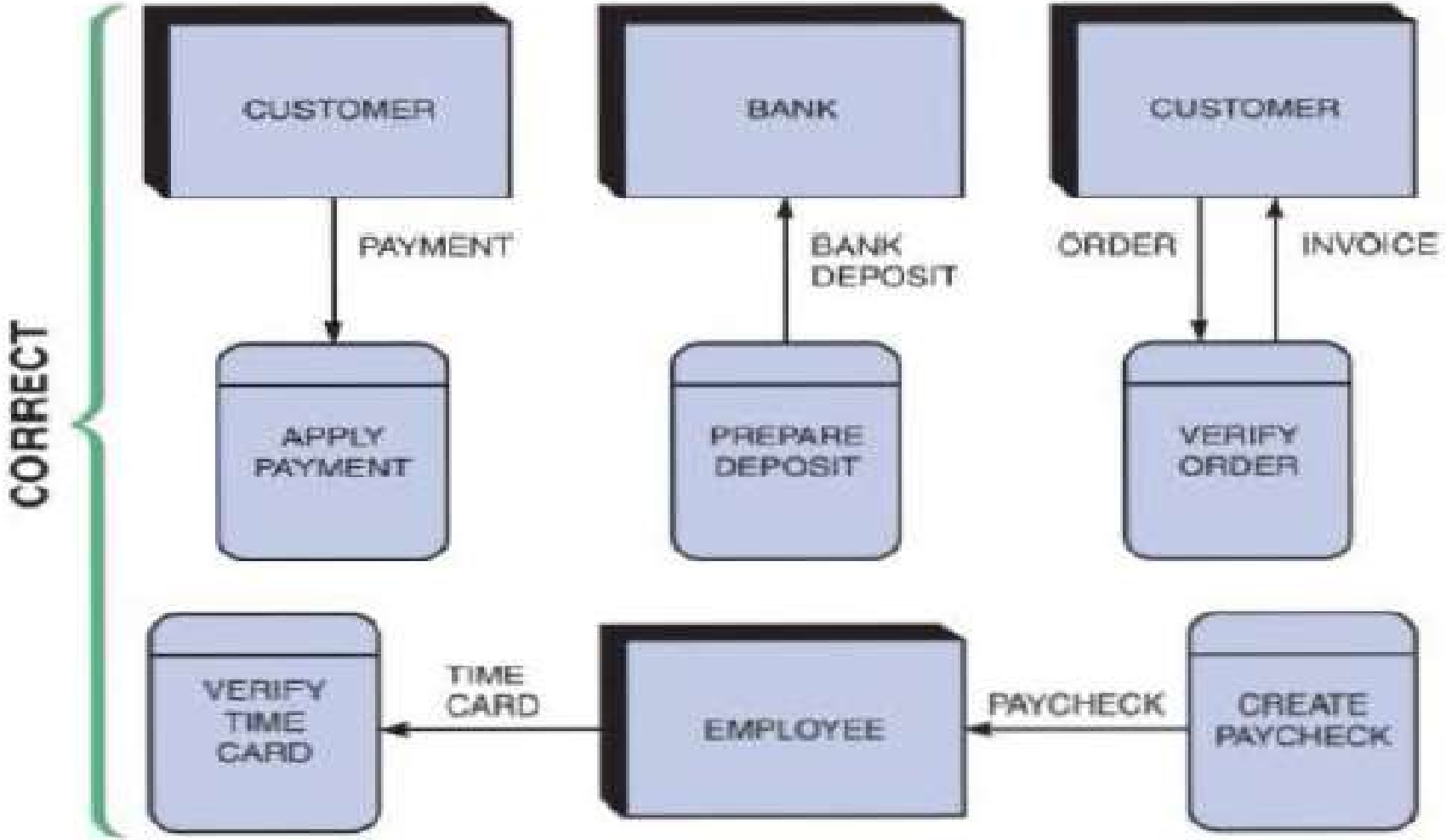


Entity Symbols (2)

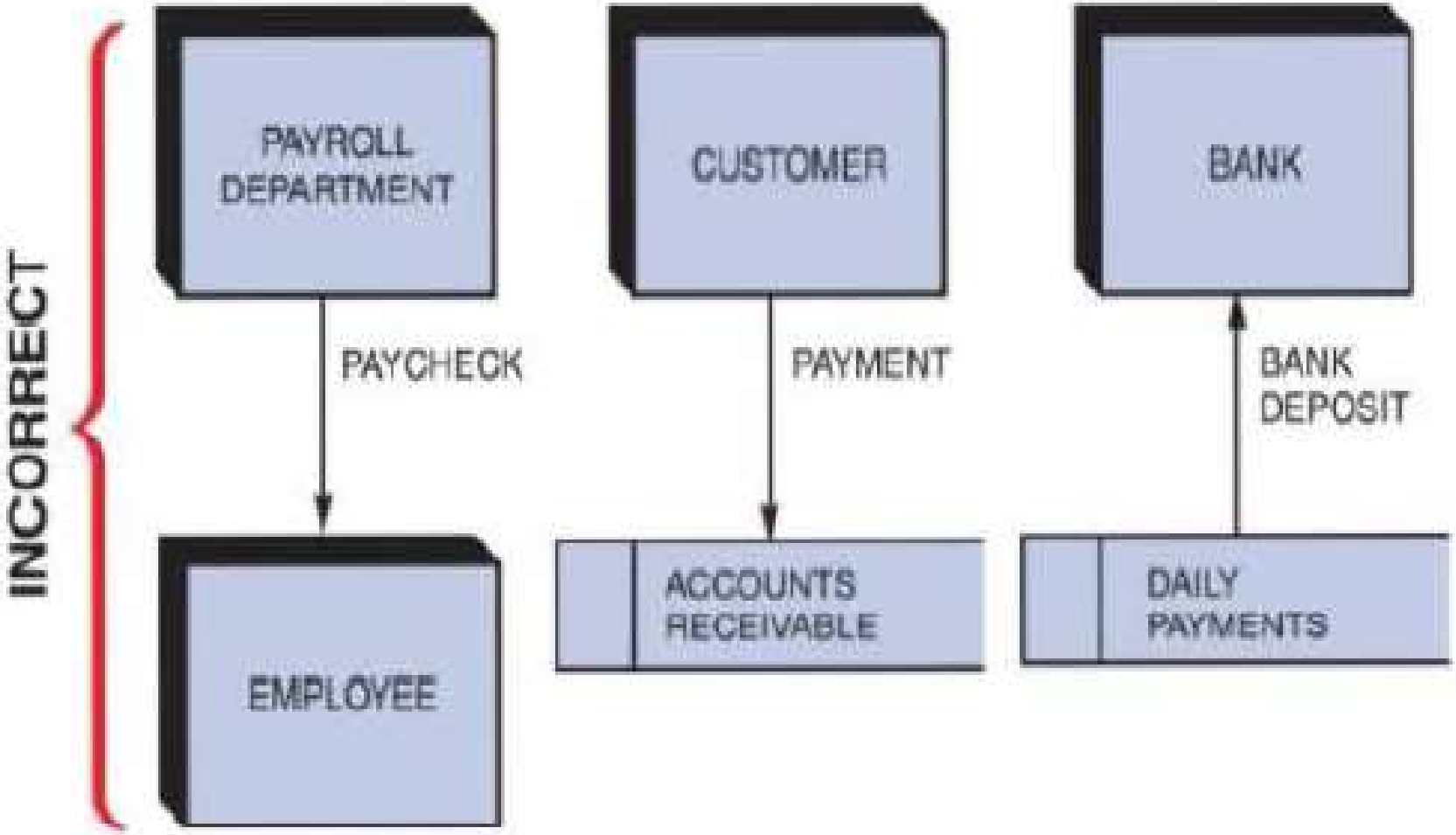
- DFD entities / **terminators** show **data origins** or **final destinations**
- An **entity supplying data** to the system is a **source** and an entity **receiving data** from the system a **sink**
- An entity **name** is the **singular form** of a department, an outside organization, other information system, or a person
- An external entity can be a **source** or a **sink** or both, but each entity **must be** connected to a **process** by a **data flow**



Entity Symbols (3)

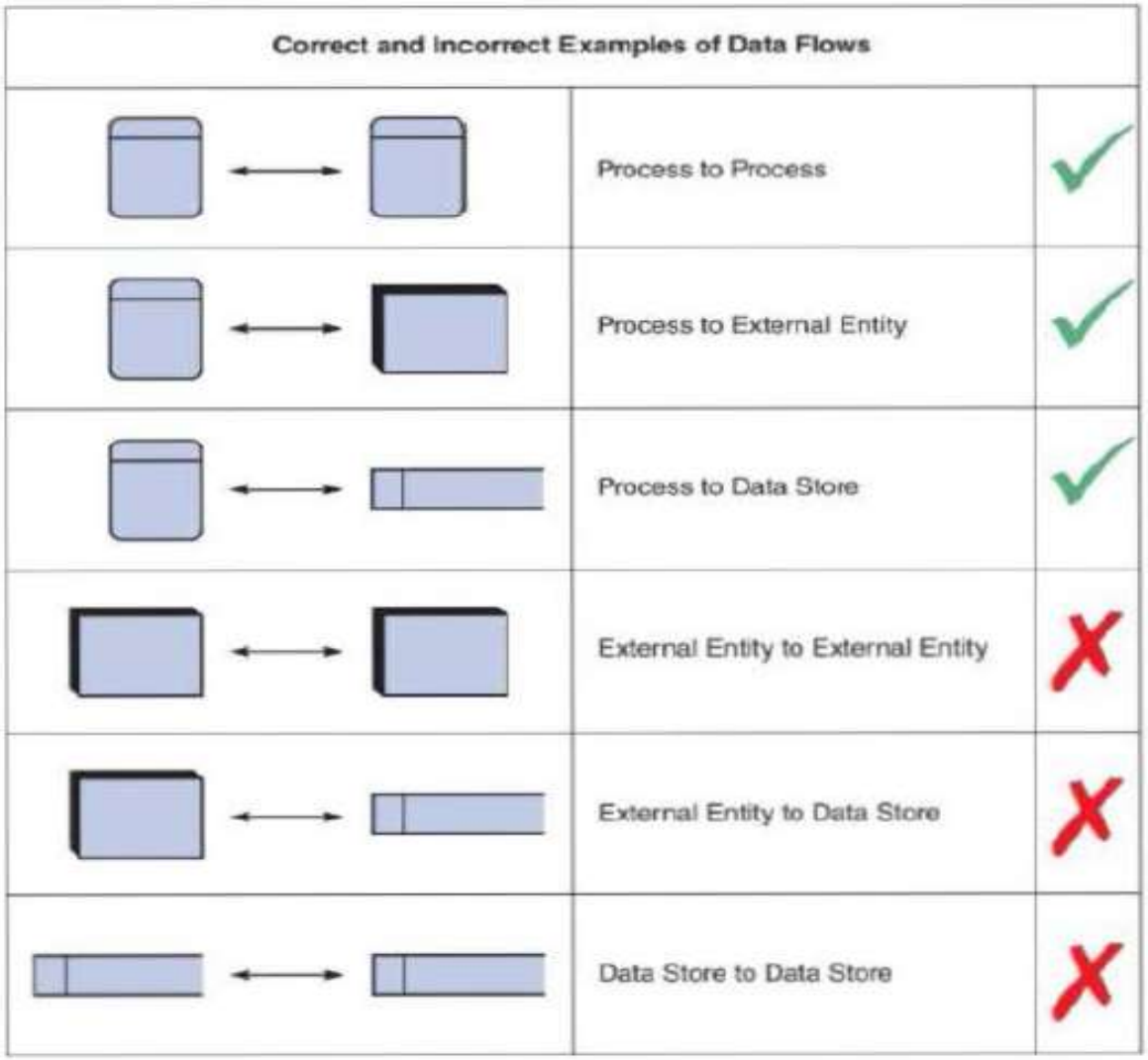


Entity Symbols (4)



Using DFD Symbols

- Summary of the **Rules for DFD Symbols**



Drawing Data Flow Diagrams

- **Guidelines** to be followed include:
  - Draw the context diagram so it **fits on one page**
  - Use the **name of the information system** as the **process name** in the context diagram
  - Use **unique names** within each set of symbols
  - **Do not cross lines**. On lower-level diagrams with multiple processes, there should *not be more than nine process symbols*
  - Provide a **unique name** and **reference number** for each **process**
  - **Obtain** as much **user input** and **feedback** as possible

Context Diagrams

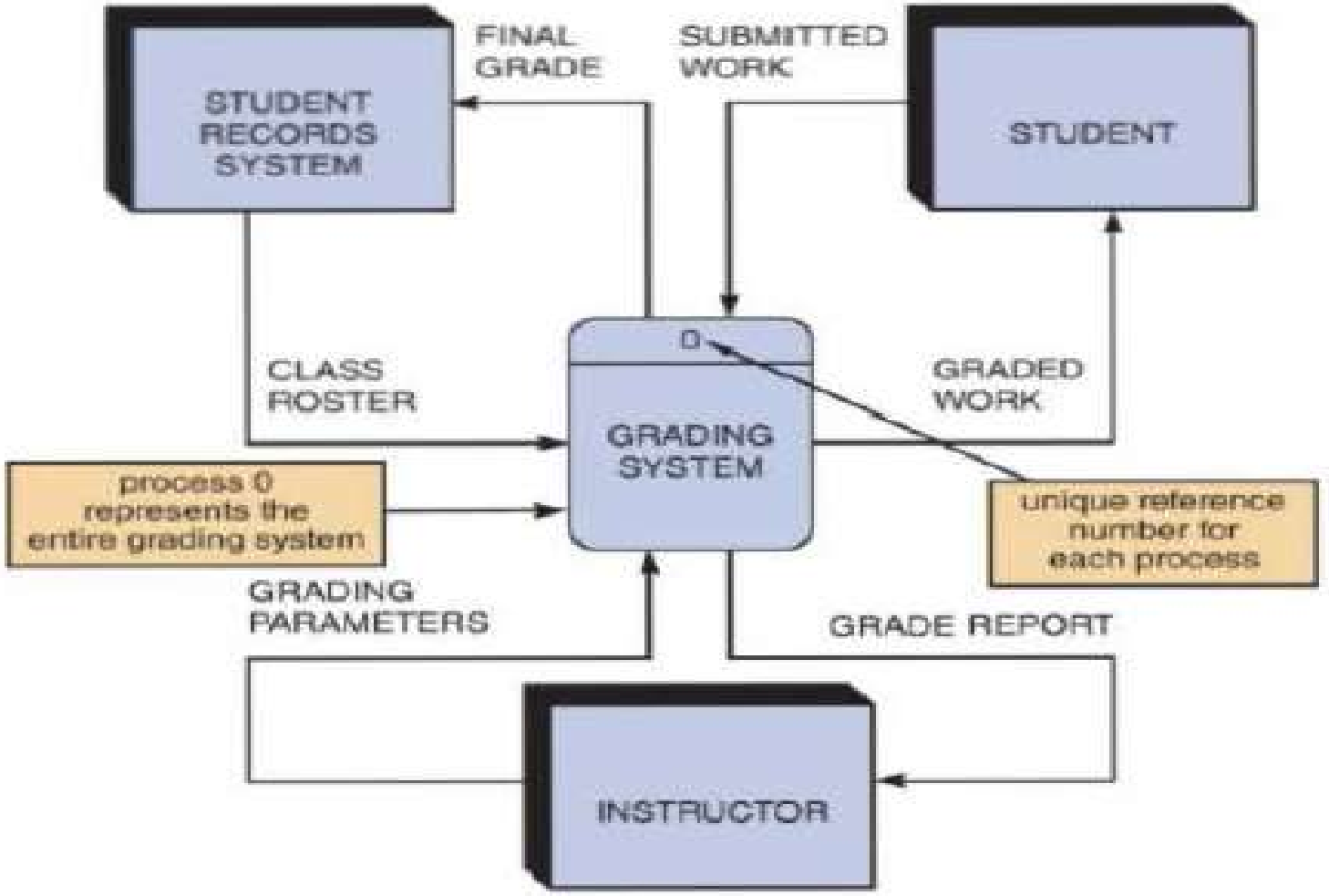
Drawing a Context Diagram

- The **first step in constructing a set** of DFDs is to draw a context diagram
- A context diagram is **a top-level view** of an **information system** that shows the system's boundaries and scope
- To draw a context diagram, start by **placing a single process symbol** in the **center** of the page
- The symbol represents the **entire information system**, and it is identified as **process 0**
- Then, **place** the **system entities** around the *perimeter of the page* and use **data flows** to **connect** the **entities** to the central process
- ***Data stores are not shown in the context diagram*** because they are contained within the system and remain hidden until more detailed diagrams are created

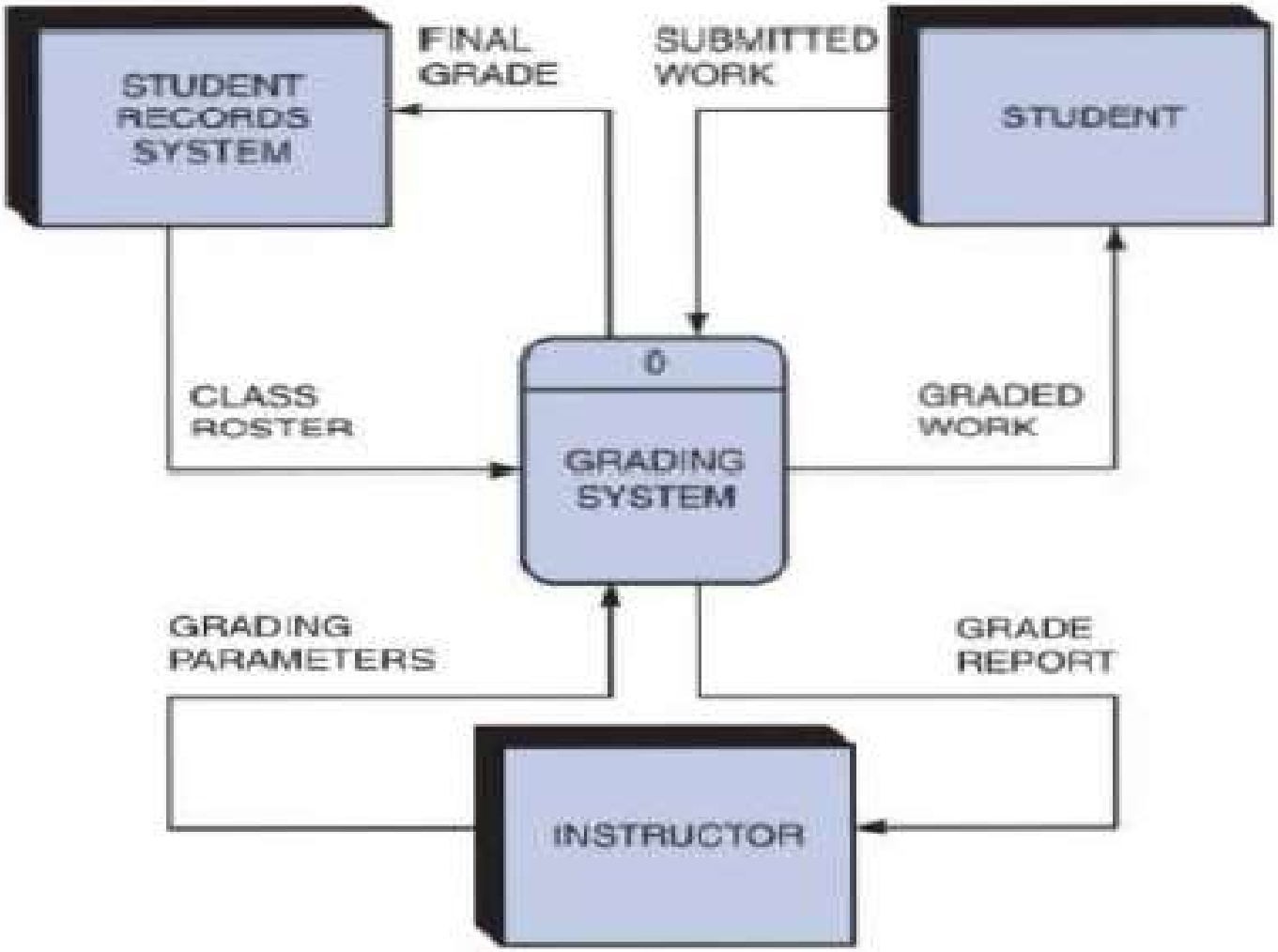
Example 1: A Grading System

- The first example is a grading system that instructors use to assign final grades based on the scores that students receive during the term:
  - The grading system process consists of the students records system, student and instructor which communicate with the grading system. Interaction among the grading system and the entities involves six different data flows. The student records system supplies data through the class register data flow and receives data through the final grade data flow. The student supplies data through the submitted work data flow and receives data through the graded work data flow. Finally, the instructor supplies data through the grading parameters data flow and receives data through the grade report data flow

Example 1: A Grading System (2)



Example 1: A Grading System (3)



Example 1: An Ordering System

- The second example is an order system that a company uses to enter orders and apply payments against a customer's balance
  - The order system process contains five entities that surround the main process. There are three entities, sales rep, bank, and accounting, whose incoming data flows are commission, bank deposit, and cash receipts entry, respectively. The warehouse entity receives data from a picking list, that is, a report chat shows the items ordered and their quantity, location, and sequence to pick from the warehouse. The warehouse entity is used to populate the completed order list. Finally, the customer entity is used to populate the order and payment, and it receives data from the order reject notice and invoice

Example 1: An Ordering System (2)

- Try it out during your free time!

- No need to submit, just for more practice…

Drawing a diagram 0 DFD

- A context diagram **provides** the **most general view** of an information system and contains a single process symbol, the black box
- To **show the detail inside** the black box, a **DFD diagram 0** is created
- Diagram 0 provides an **overview** of all the **components** that **interact** to form the **overall system**
- It **zooms in on the system** and shows *major internal processes*, *data flows*, and *data stores*

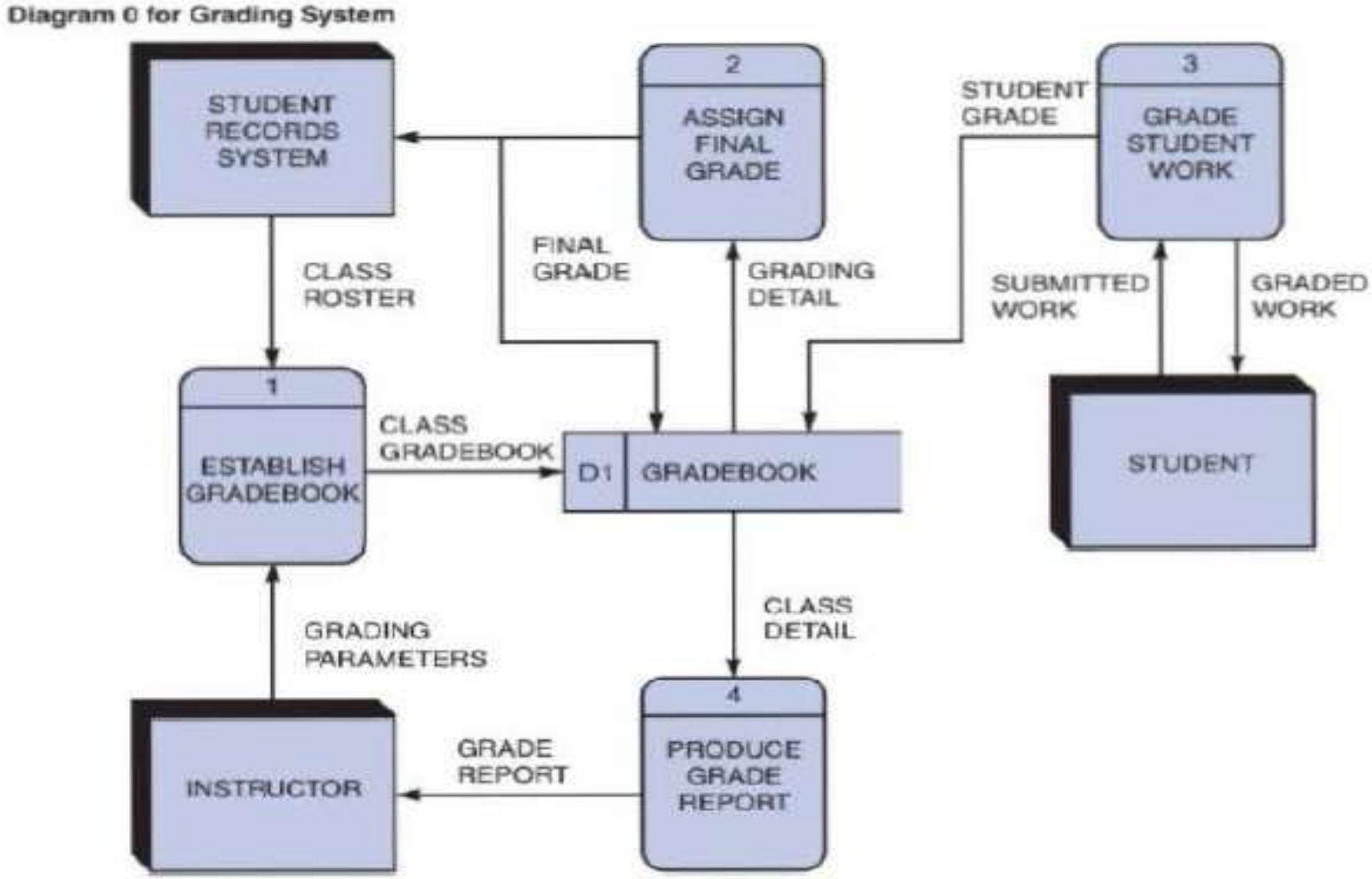
Drawing a diagram 0 DFD (2)

- **Diagram 0** expands **Process 0**

- Also note that the **three same entities** (STUDENT RECORDS SYSTEM, STUDENT, and INSTRUCTOR) and the same six **data flows** (FINAL GRADE, CLASS ROSTER, SUBMITTED WORK, GRADED WORK, GRADING PARAMETERS, and GRADE REPORT) **appear in Diagram 0 as well**

- In addition, **diagram 0 expands process 0** to reveal **other internal processes**, **a data store**, **additional data flows**

Drawing a diagram 0 DFD (3)



Drawing a diagram 0 DFD (4)

- **Each process** in diagram 0 has a **reference number**
- The **process numbers do not suggest** that the processes are accomplished in a **sequential order**
- There can be **a diverging data flow** in which the **same data travels** to *two* or *more* different **locations**
- If the **same data flows** in both directions, a **double-headed arrow** can be used to connect the symbols

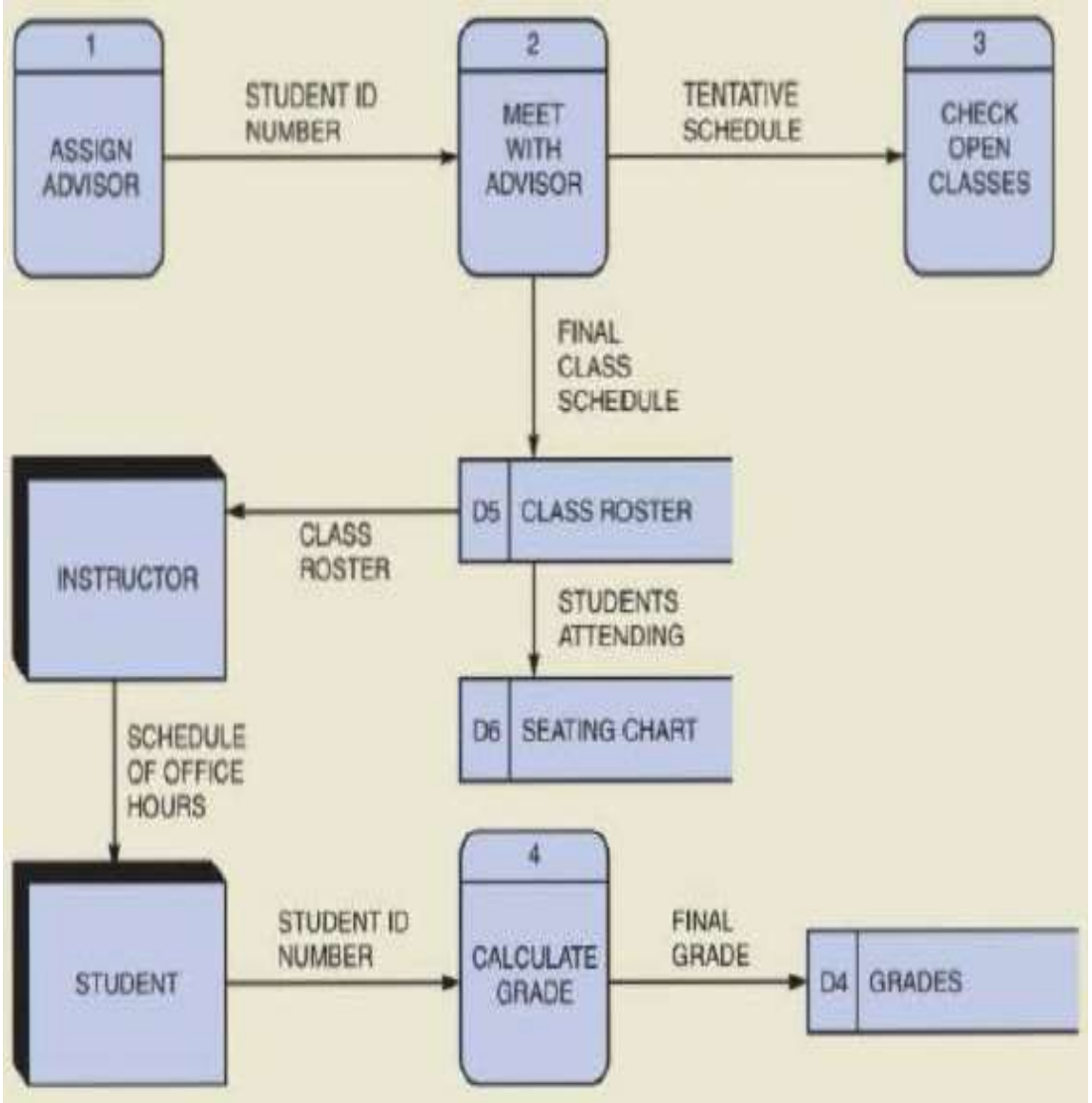
Other Terminology Used in Drawing a Diagram 0 DFD (5)

- **A functional primitive**
  - is a process that consists of a **single function** that is **not exploded further**
- **Leveling/exploding/partitioning/decomposing**
  - is the process of **drawing** a series of increasingly detailed diagrams, **until** all **functional primitives** are **identified**
- **Balancing**
  - maintains consistency on parent and child DFDs by ensuring that **input** and **output data flows align properly**

Exercise!!!

You are the ICT director at SU. As part of a training program, you decide to draw a DFD that includes some obvious mistakes to see whether your newly hired analysts can find them. You came up with the diagram 0 DFD shown in this figure.

- Based on the rules explained in this lecture, how many problems should the analysts find?



Data Dictionary

- This is **a central storehouse of information** about the system's data
- An analyst uses the data dictionary to **collect**, **document**, and **organize specific facts** about the system, including the contents of data flows, data stores, entities, and processes
- The data dictionary **defines** and **describes** all **data elements** and **meaningful combinations** of data elements

Data Dictionary (2)

- A data element/ a data item/ field is the **smallest piece of data** that has **meaning within** an **information system**

- Data elements are **combined** into **records/data structures**

- A record is a **meaningful combination** of **related data elements** that is included in a data flow or retained in a data score