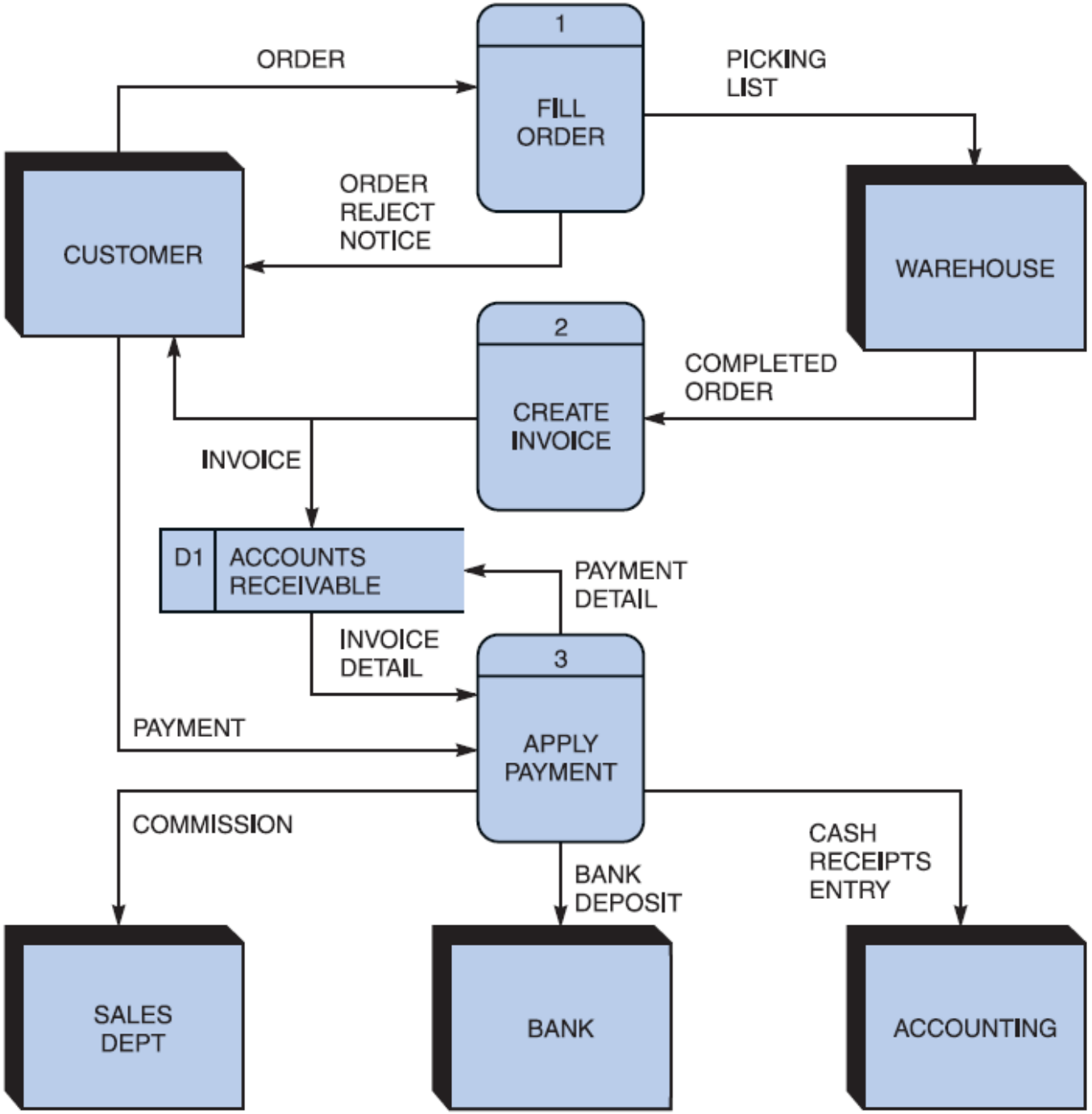
# Software Engineering

## Lecture 6

| Lectures | Topics |
|---|---|
| 1 | Introduction to Software Engineering |
| 2 | Software Development Process (SDLC Activities)<br>- SDLC Activity: Specification or Requirement Engineering<br>- SDLC Activity: System Modeling/Design<br>- SDLC Activity: Implementation<br>- SDLC Activity: Testing<br>- SDLC Activity: Evolution<br>- SDLC Activity: Deployment/Installation<br>- SDLC Activity: Maintenance |
| 3 | SDLC Activity: Requirement Engineering<br>- Requirement Elicitation<br>- Requirement Analysis and Management<br>- Requirement Validation |
| 4, 5, 6 | SDLC Activity: System Modeling/Design<br>- Context Modeling<br>- Data Modeling<br>- Structural/Architectural Modeling<br>- Process Modeling<br>- UI/UX Modeling |
| 7,8,9 | SDLC Activity: Implementation (Coding, tools, GIT – Version management, IDE, RESTFUL architecture) |
| 10 | SDLC Activity: Testing |
| 11 | SDLC Activity: Deployment (tools to deploy, cloud computing) |
| 12 | SDLC Activity: Maintenance |

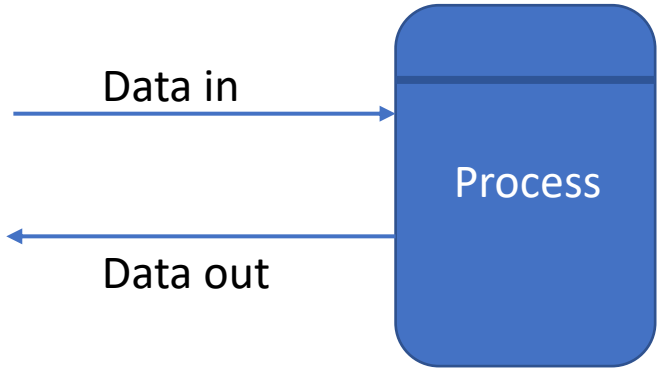
# Structural Process Modeling


1
FILL ORDER
ORDER
PICKING LIST
CUSTOMER
ORDER REJECT NOTICE
WAREHOUSE
2
CREATE INVOICE
COMPLETED ORDER
INVOICE
D1
ACCOUNTS RECEIVABLE
PAYMENT DETAIL
INVOICE DETAIL
3
APPLY PAYMENT
PAYMENT
COMMISSION
CASH RECEIPTS ENTRY
BANK DEPOSIT
SALES DEPT
BANK
ACCOUNTING

# DFD Diagram

# 1.0 Data Flow Diagram

- A DFD shows HOW data moves through an information system but does not show program logic or processing steps.
- A set of DFDs provides a logical model that shows WHAT the system does, NOT how it does it



1.1

# DFD Symbols (Gane and Sarson)

**We will be using Gane and Sarson Symbols**

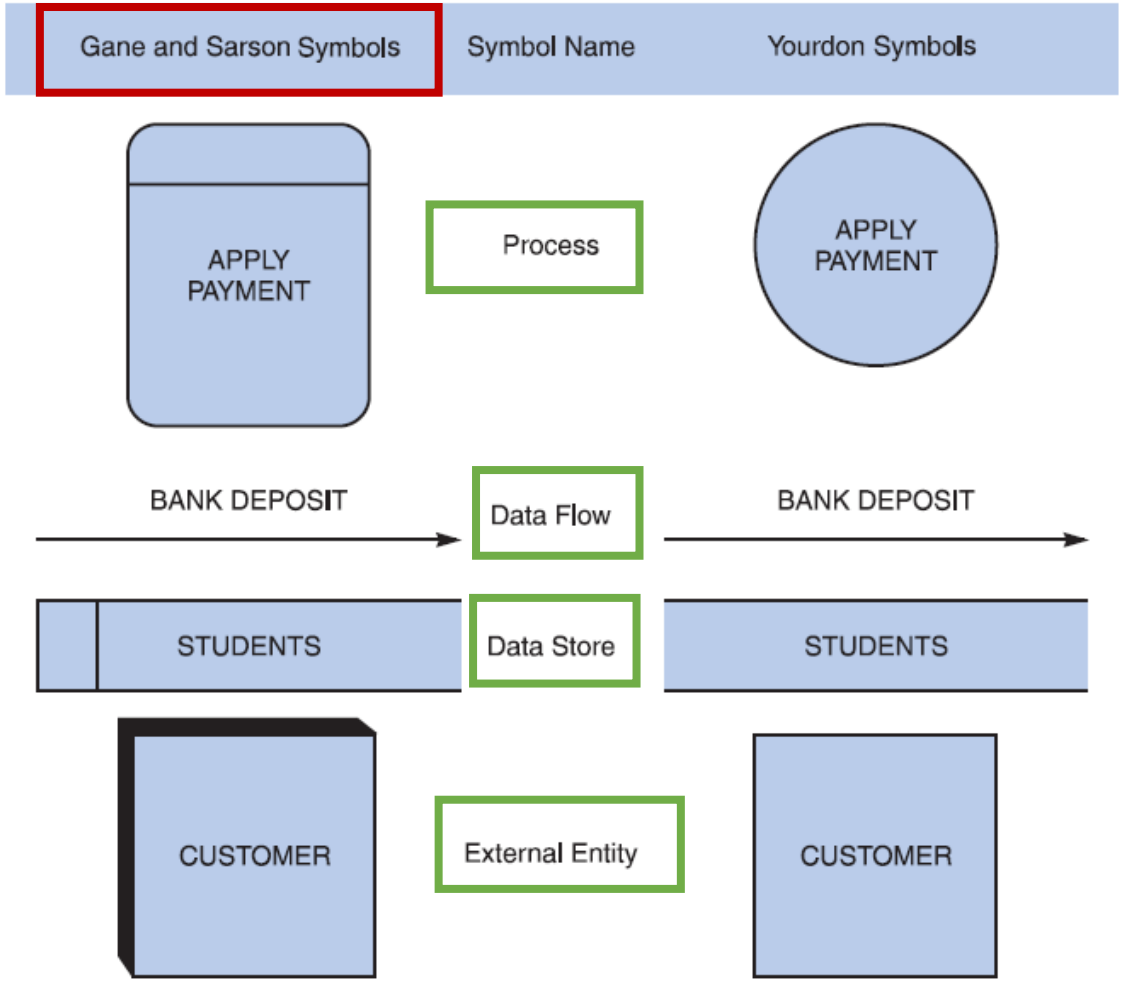
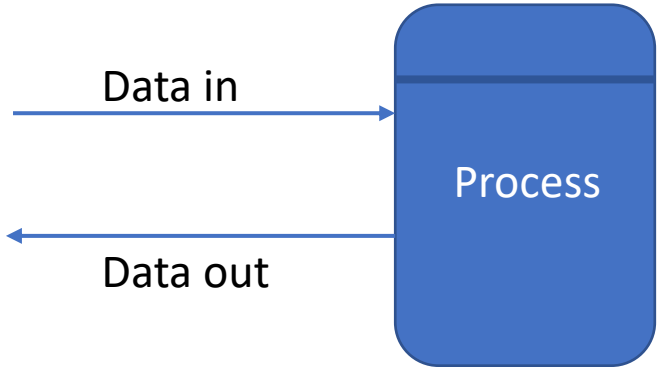


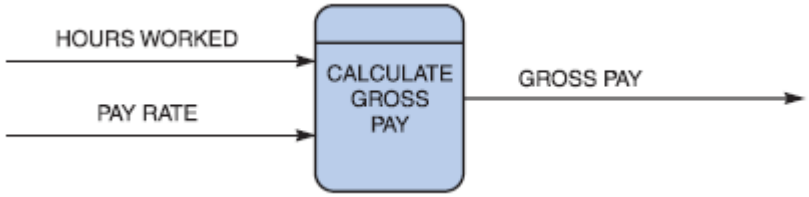

**FIGURE 5-3** Data flow diagram symbols, symbol names, and examples of the Gane and Sarson and Yourdon symbol sets.

# 1.1.1 DFD Symbol: Process Symbol




- Process represent the business logic / rules but does not reveal the detail steps. Process is also sometimes called ‘black-box’ (Analogy: Router with cable attached to it)
- Process transform the data and produces the required results
- 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.
- **Naming convention**: verb followed by a singular noun [apply payment]

# 1.1.2 DFD Symbol: Data Flow Symbol




- Is a path for data to move from one part of the information system to another
- A data flow could consist of a **single data item** (such as a student ID number) or it could include a **set of data** (such as a class roster with student ID numbers, names, and registration dates for a specific class)
- Naming Convention: consist of a singular noun and an adjective if needed [Deposit, Invoice payment, student grade, order, commission]
- The data flow name appears above, below, or alongside the line.

# 1.1.3 DFD Symbol: Data Store Symbol


- To represent data that the system stores
- A DFD does not show the detailed contents of a data store
- Process might need access to the data
- 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 adjectives, if needed. Examples of data store names are STUDENTS, ACCOUNTS RECEIVABLE, PRODUCTS, DAILY PAYMENTS, PURCHASE ORDERS, OUTSTANDING CHECKS, INSURANCE POLICIES, and EMPLOYEES.

1.1.4

# DFD Symbol: Entity Symbol


- Are also called terminators
- Provide data to the system or receive output from the system.
- Entities also are called terminators, because they are data origins or final destinations
- A rectangle, which may be shaded to make it look three-dimensional.
- Systems analysts call an entity that supplies data to the system a **source**, and an entity that receives data from the system a **sink**

1.2

# Correct way for drawing DFDs

LAB

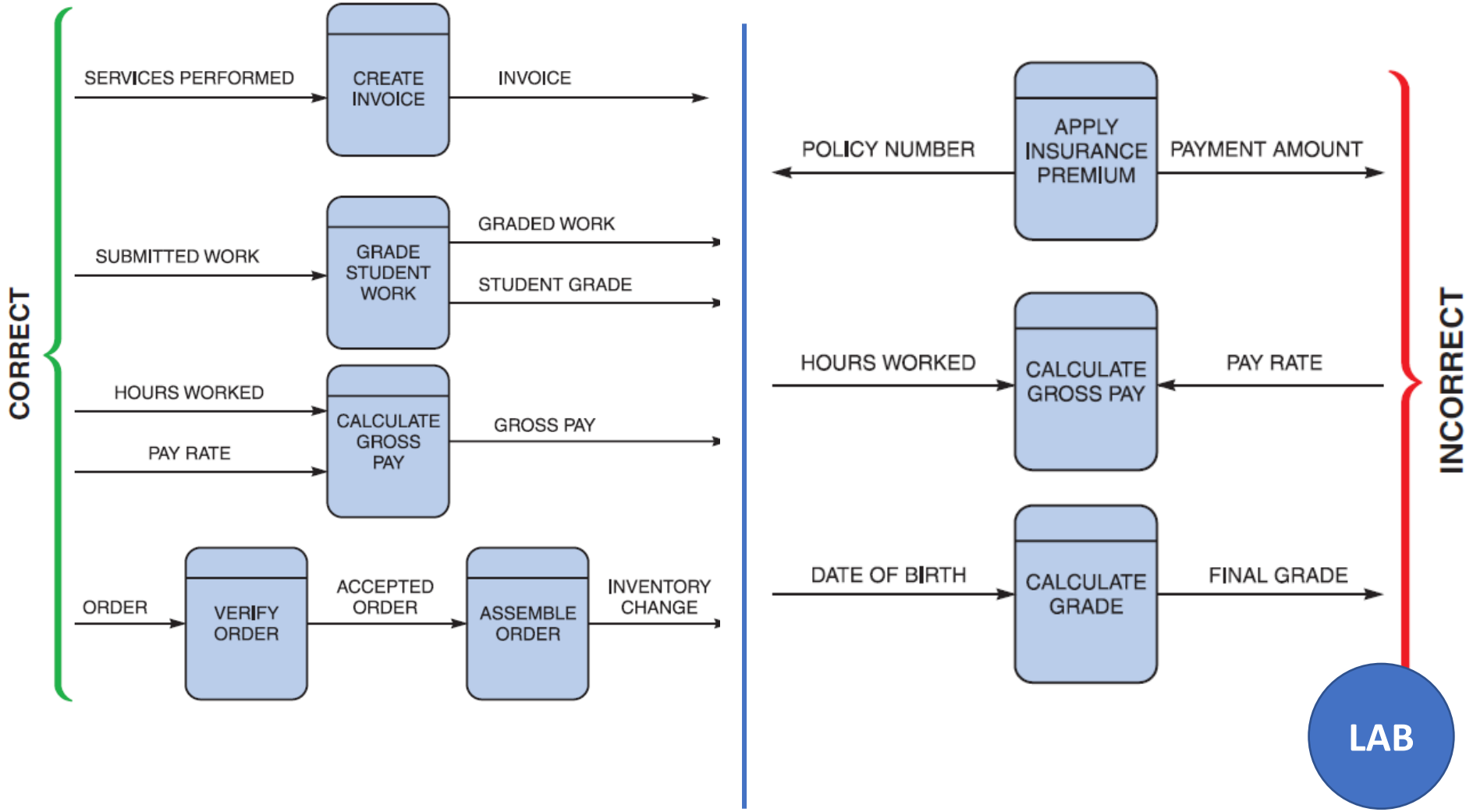
# 1.2.1 Correct combinations of data flow and process

- Process symbol can have more than one outgoing data flow.
- Process can connect to any other DFD symbol including process symbol
- Three Data flow and process combination that we must avoid
  - Spontaneous Generation
    - Process producing output without input (exception applies)
  - Black Hole
    - Process with input, but produces no output (exception applies)
  - Gray Hole
    - Gray hole is a process that has at least one input and one output, but the input obviously is insufficient to generate the output

LAB

1.2.1

# Correct combinations of data flow and process



1.2.2

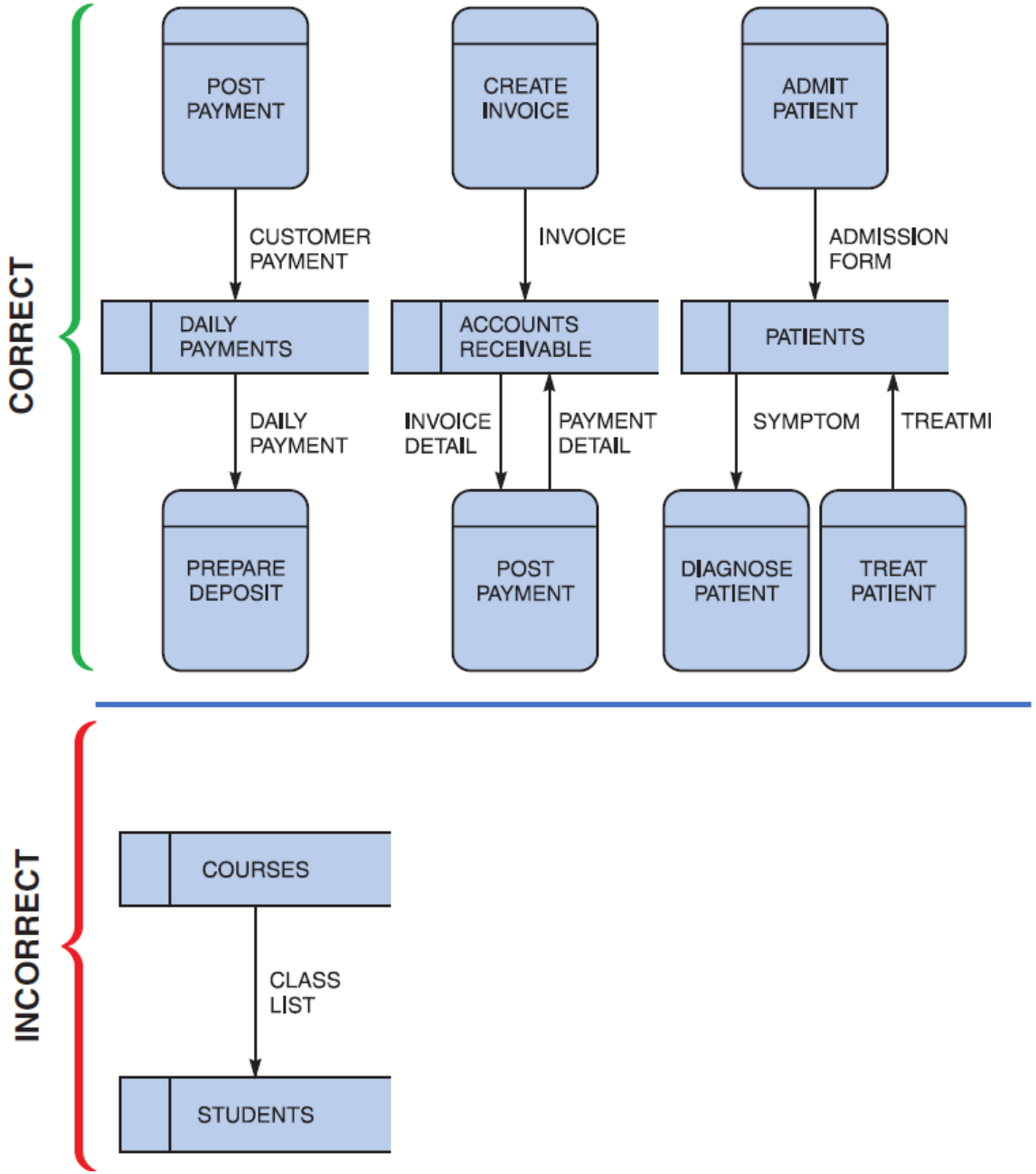
# Correct uses of data store

A data store must be connected to a process with a data flow

LAB

1.2.2

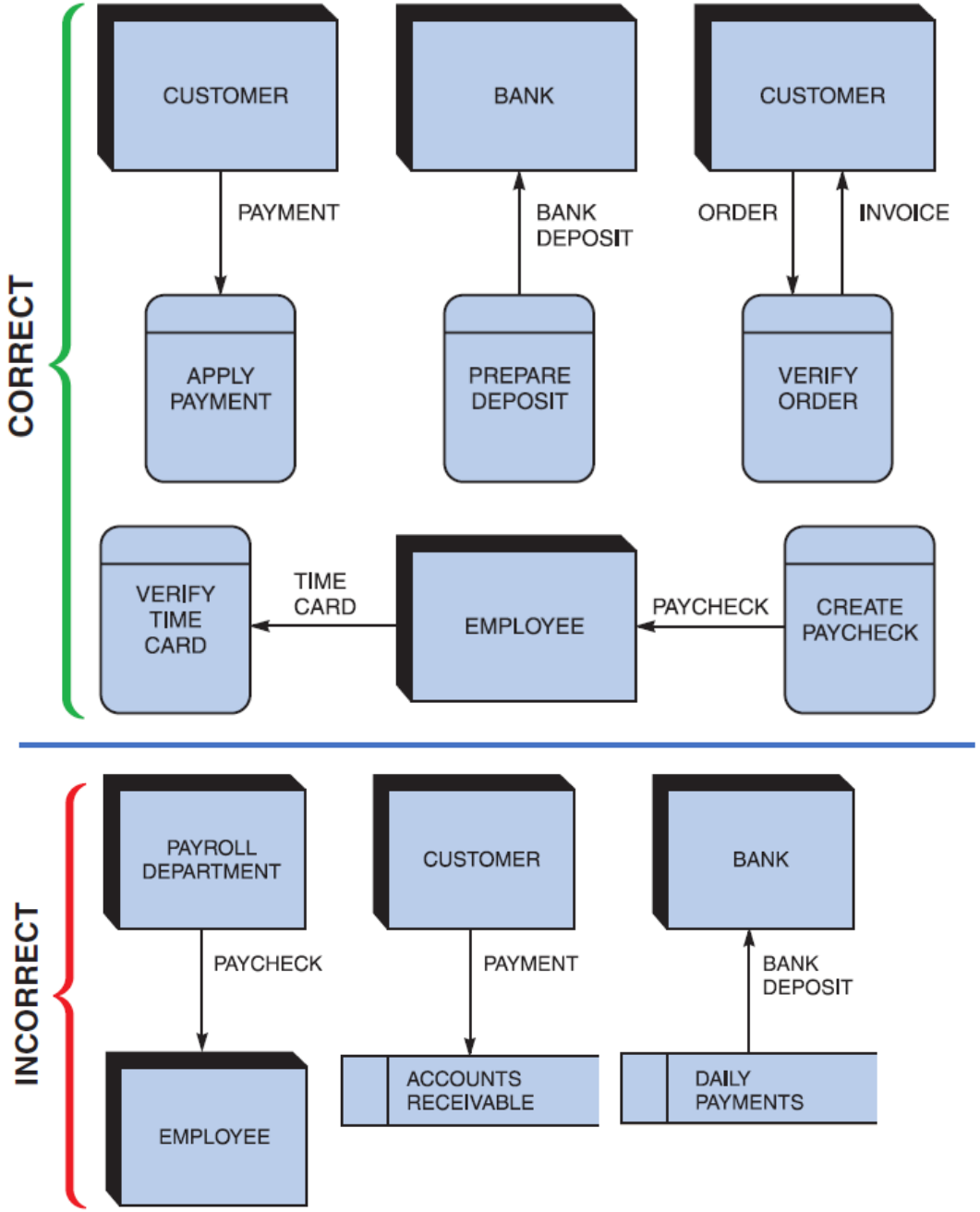
# Correct uses of data store




LAB

1.2.3

# Correct uses of external Entities

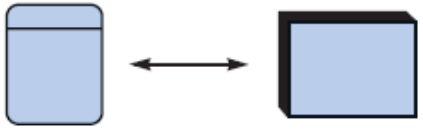
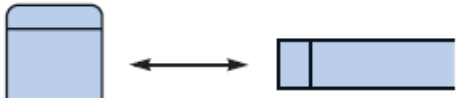
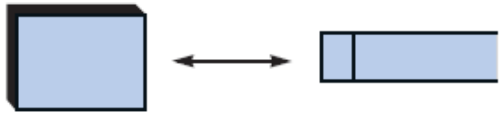



LAB

1.2.4

General summary of correct usage of symbol

LAB

| Correct and Incorrect Examples of Data Flows | | |
|---|---|---|
|  | Process to Process | ✓ |
|  | Process to External Entity | ✓ |
|  | Process to Data Store | ✓ |
|  | External Entity to External Entity | ✗ |
|  | External Entity to Data Store | ✗ |
|  | Data Store to Data Store | ✗ |

1.3

# Steps of Drawing DFDs (Top-Down Approach)

Step 1

Context Diagram

Step 2

Upper level Diagram

Step 3

Lower Level Diagram

- Note:
  - Do not cross the line (data flow)
    - A way to avoid too many crossing lines is to duplicate an entity or data store. When duplicating, put asterisk as a symbol in duplicating entities or data store.
  - On lower level diagram with multiple processes, you should not have more than nine process symbols (a good practice). Including more than nine process symbols usually is a signal that your diagram is too complex and that you should reconsider your analysis.

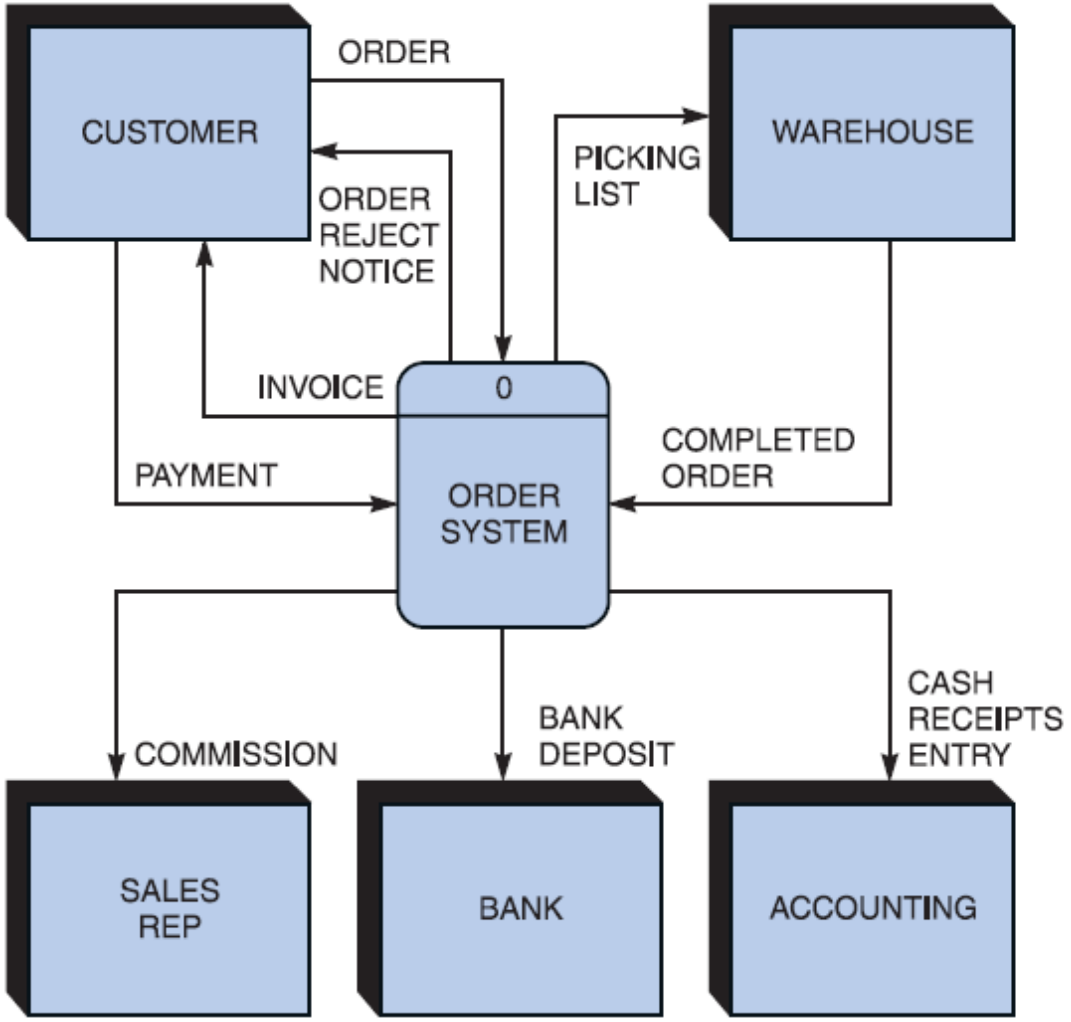
LAB

- Before Going to each steps, let's hypothesize a system on which we will carry out the different steps involved in drawing DFD in top-down approach
- **Order System**
  - To create customer order
  - Send invoice of order to customer
  - Customer can make payment against the invoice

LAB

1.3.1

# Step 1: Context Diagram

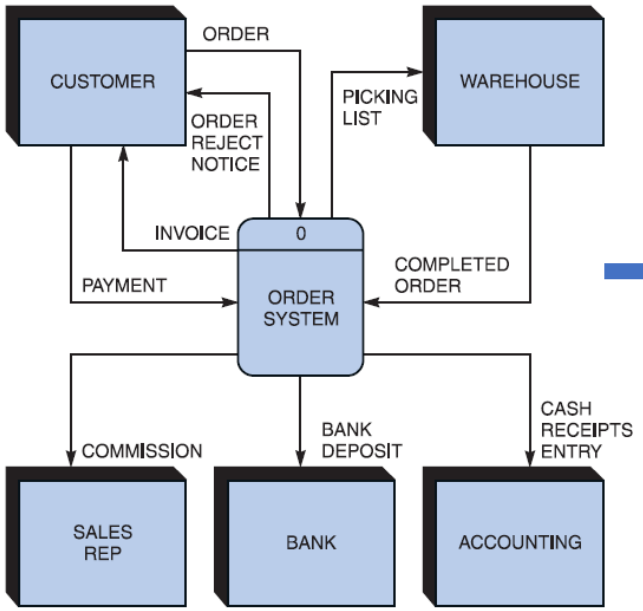



LAB

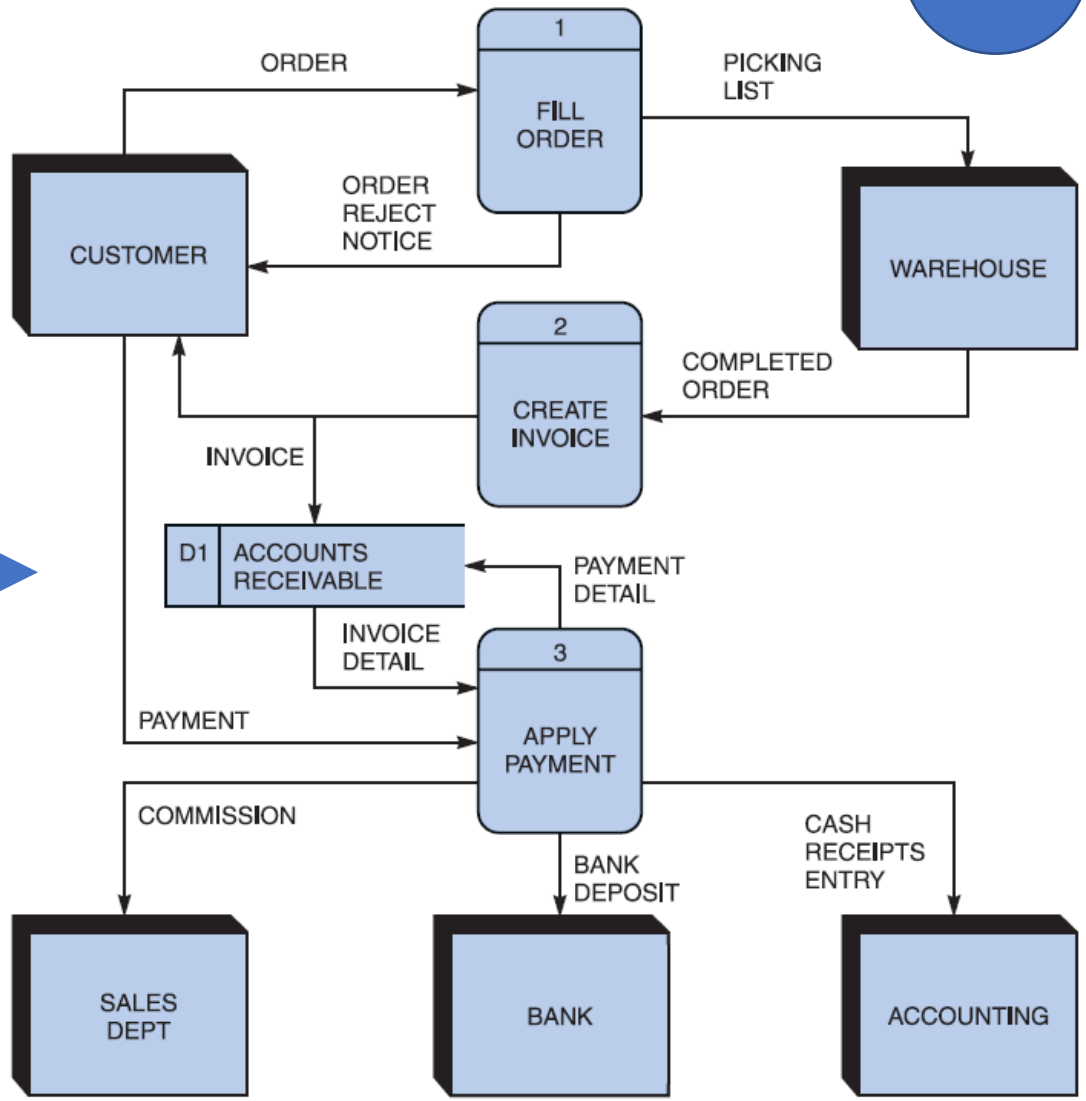
1.3.2

# Step 2: Upper level Diagram

LAB

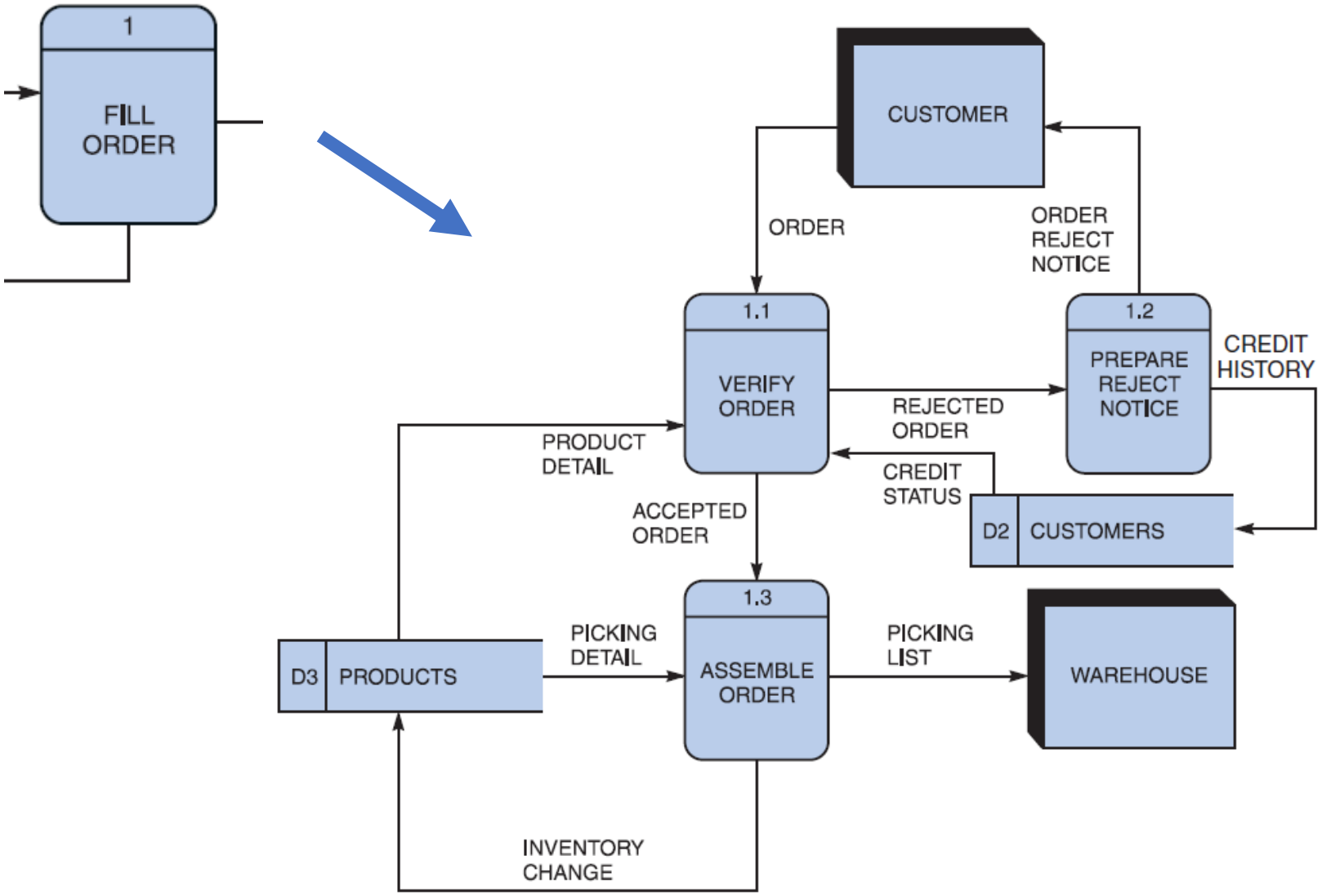



Context Diagram




Upper Level Diagram

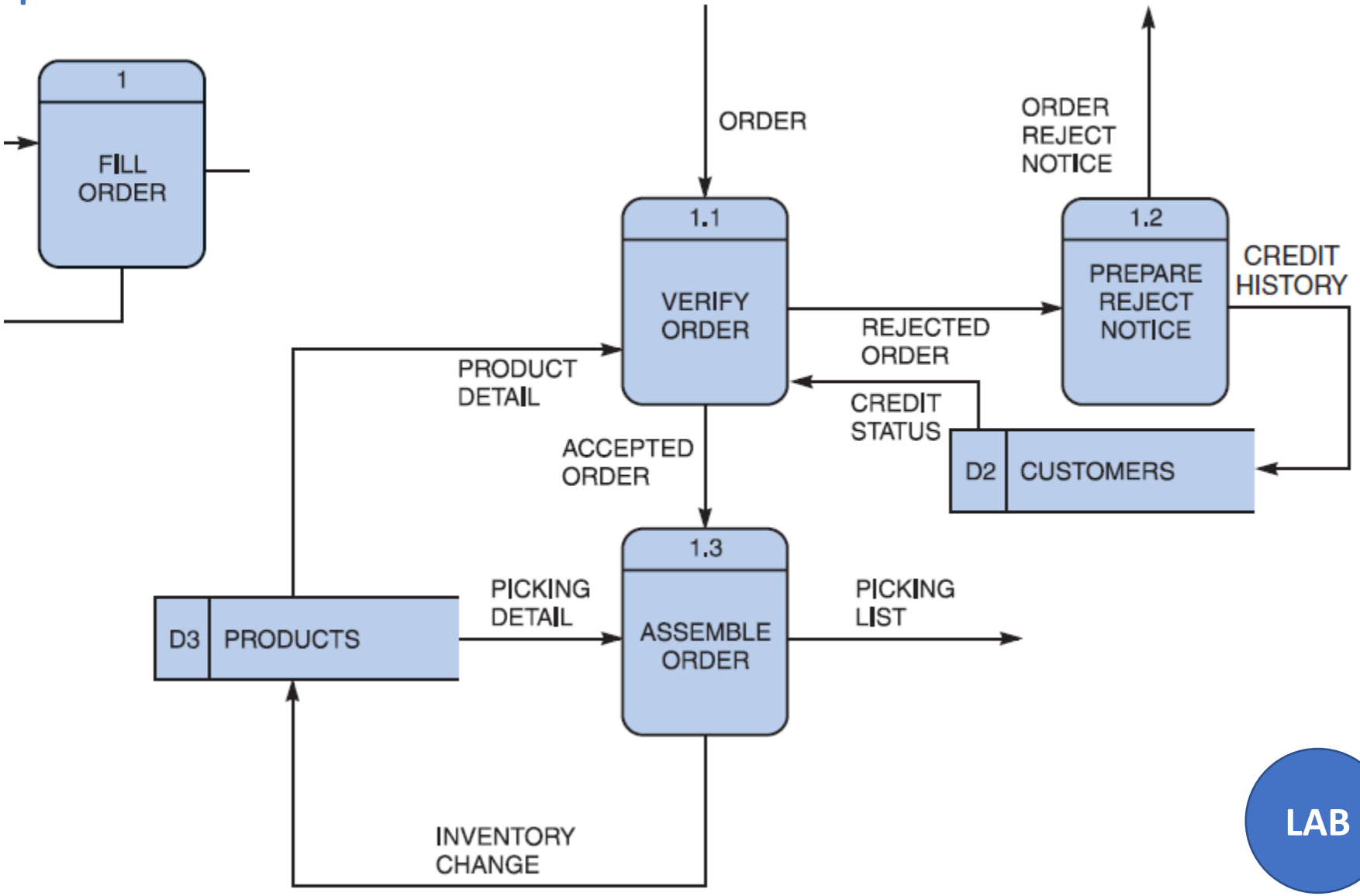
# 1.3.3 Step 3: Lower Level Diagram




LAB

1.3.3

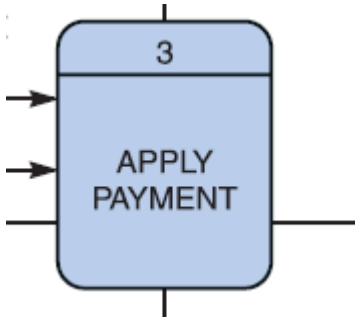
# Step 3: Lower Level Diagram (leveling out)

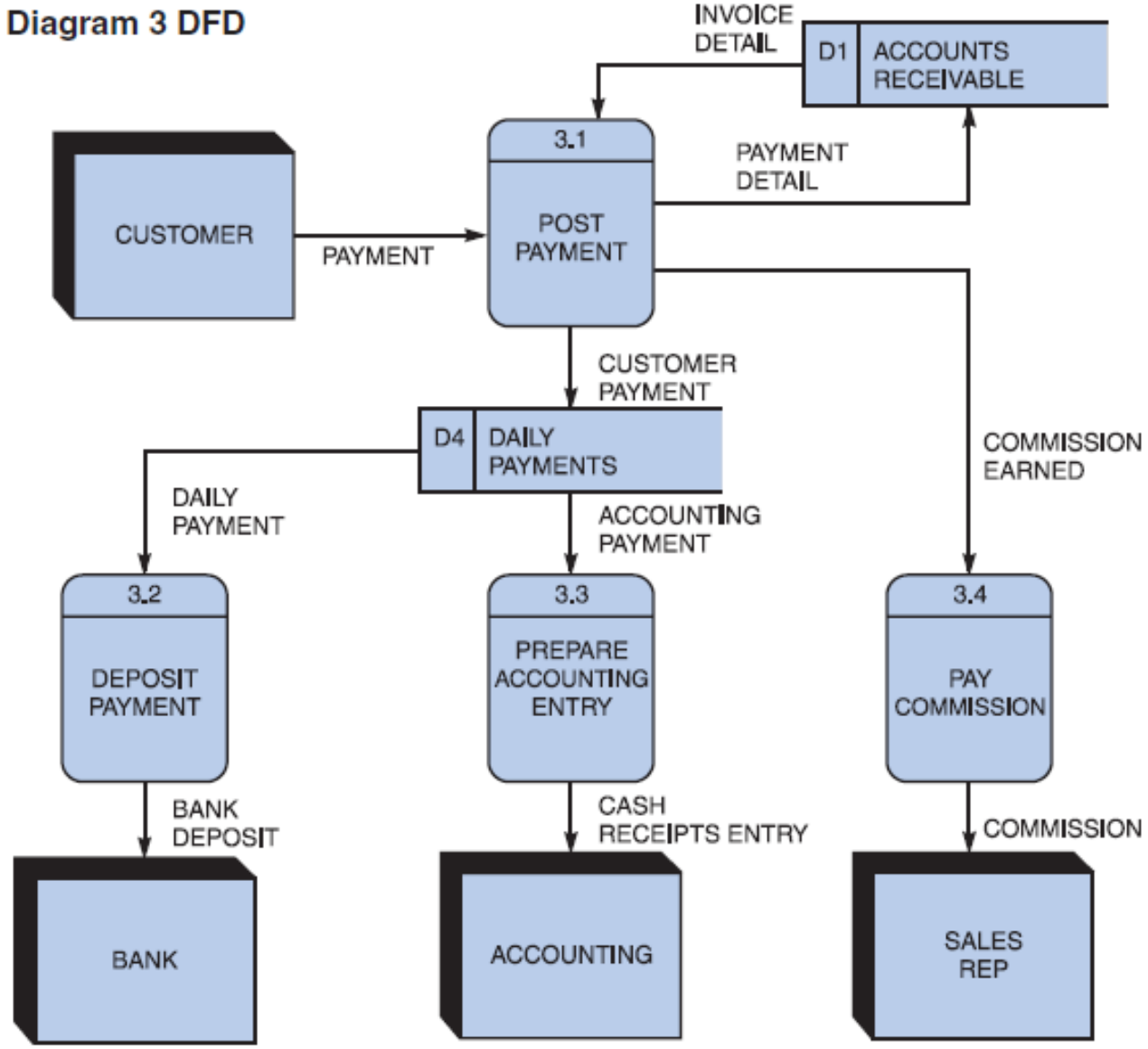



LAB

1.3.3

# Step 3: Lower Level Diagram




LAB

# 2.0 Process Description Tool

- Process in lower level DFD are still not in detail and do not exactly explain the actual flows of how things works inside the process. This is where, we use different process description tool to describe the flow of sequence inside the process. Also knows as PSPECS

# 2.0 Process Description Tool

**To describe the process or flow, we generally use two ways**

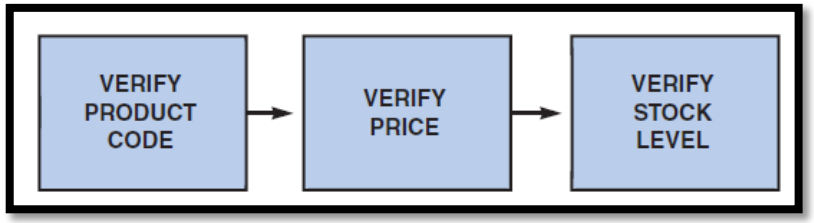
1. Modular Design (Flow chart)
2. Structured English (Algorithm, Pseudo Code)
3. Decision Tables (logical table)
4. Decision Tree (Decision Branching)

**Regardless of whatever tools we use (above two tools), we use three different logical structures which serves as building blocks**

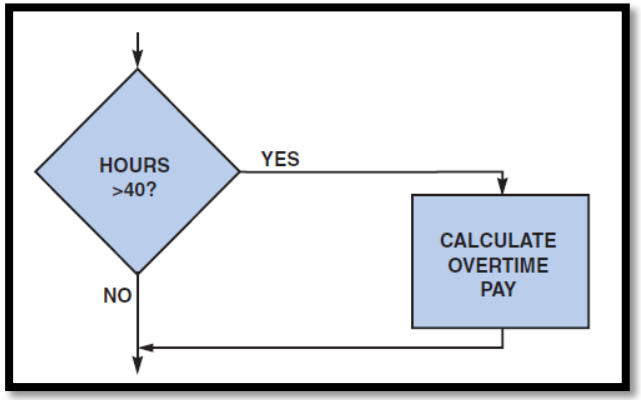
1. Sequence (one after another, linear flow)
2. Selection (alternative paths, if-else scenario)
3. Iteration (looping scenario)

# 2.1 Modular Design

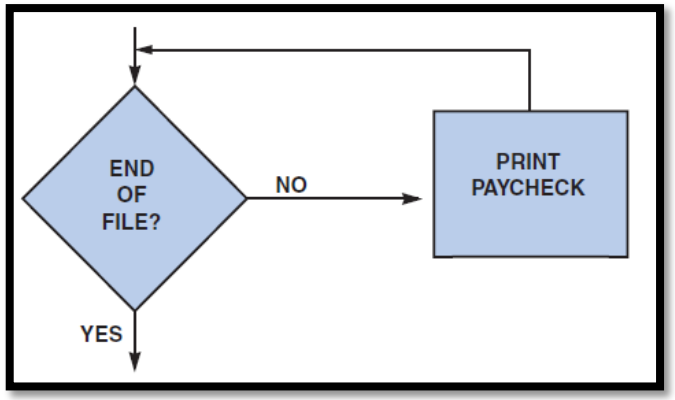
- Flow Chart is an example of Modular Design tool




Logical Structure- Sequence




Logical Structure - Iteration




Logical Structure- Selection

2.2

# Structured English

- Example tool: Algorithm, Pseudo-code
- Make the use of indentation and limited vocabulary
- Also make use of logical structure (sequence, selection and iteration)

```
If (marks > 50)
        Declare pass
Else
        Declare fail
```

# 2.3 Decision Tables

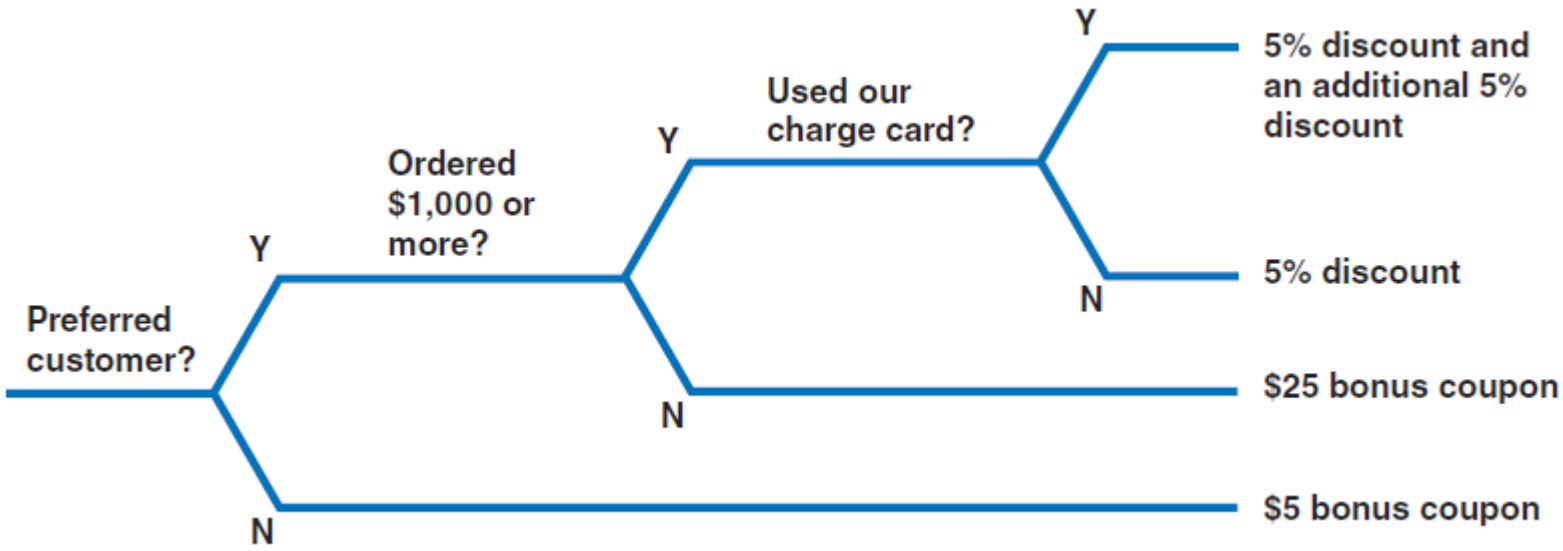
Is a logical math like table

| VERIFY ORDER Business Process with Two Conditions |
| --- |
| • An order will be accepted only if the product is in stock and the customer's credit status is OK. |
| • All other orders will be rejected. |

| | 1 | 2 | 3 | 4 | |
| --- | --- | --- | --- | --- | --- |
| Credit status is OK | Y | Y | N | N | 3 |
| Product is in stock | Y | N | Y | N | |
| Accept order | X | | | | 4 |
| Reject order | | X | X | X | |

2.4

# Decision Trees



# 3.0 Data Dictionary

Data Dictionary is useful to describe the details of DFDs.

Data Dictionary can be created using CASE tools. Popular CASE tool are 'Visible Analyst'

1. Data Elements (Attributes of Data store)
2. Data Flows (content that flows over data flow)
3. Data Stores (Data holder)
4. Processes (function, representing business logic)
5. Entities (External entities interacting with system)
6. Records (set of related data elements)