

Learning Objectives

- ❖ What are Data Flow Diagrams (DFDs)?
- ❖ Why they are useful?
- ❖ How are they developed?
- ❖ How to level DFDs?
- ❖ Good style conventions in developing DFDs
- ❖ Difference between Logical and Physical DFDs
- ❖ Tools available to draw DFDs

Motivation

WHY DFD ?

Provides an overview of

- What data a system processes
- What transformations are performed
- What data are stored
- What results are produced and where they flow

Graphical nature makes it a good communication tool between

- User and analyst
- Analyst and System designer

Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams



DATA FLOW DIAGRAMS

WHAT ARE DATA FLOW DIAGRAMS?

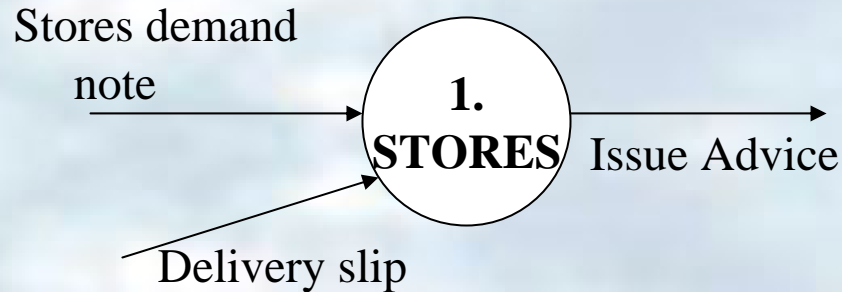
DFDs models the system by depicting

- External entities from which the data flows and where results terminate
- Processes which transform data flows
- Data stores from which the data are read or into which data are written by the processes.



SYMBOLS USED IN DFD

PROCESS



- A circle represents a process
- Straight lines with incoming arrows are input data flows
- Straight lines with outgoing arrows are output data flows
- Processes are given serial numbers for easy reference
- Labels are assigned to Data flow. These aid documentation



SYMBOLS USED IN DFD

EXTERNAL ENTITIES

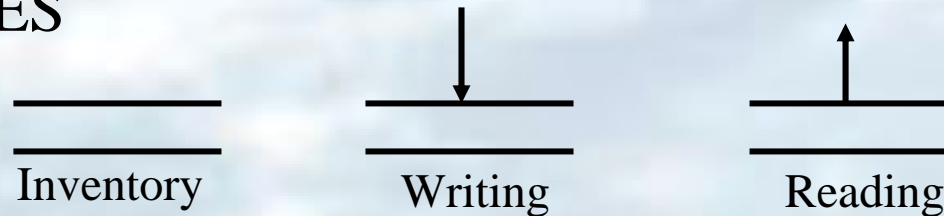


- A Rectangle represents an external entity
- They either supply data or receive data
- They do not process data



SYMBOLS USED IN DFD

DATA STORES



- A Data Store is a repository of data
- Data can be written into the data store
This is depicted by an incoming arrow
- Data can be read from a data store
This is depicted by an outgoing arrow
- External entity cannot read or write to the data store
- Two data stores cannot be connected by a data flow



RULES OF DATA FLOW

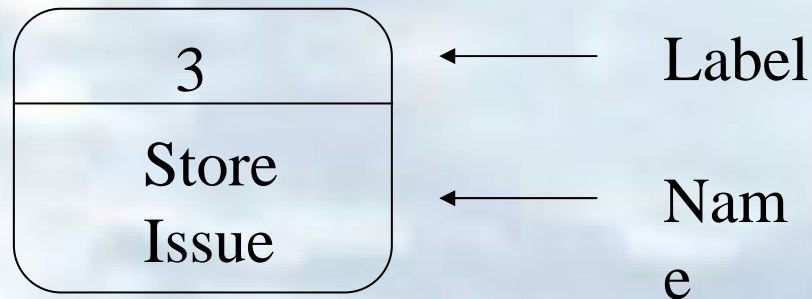
- Data can flow from
 - external entity to process
 - process to external entity
 - process to store and back
 - process to process
- Data cannot flow from
 - external entity to external entity
 - external entity to store
 - store to external entity
 - store to store



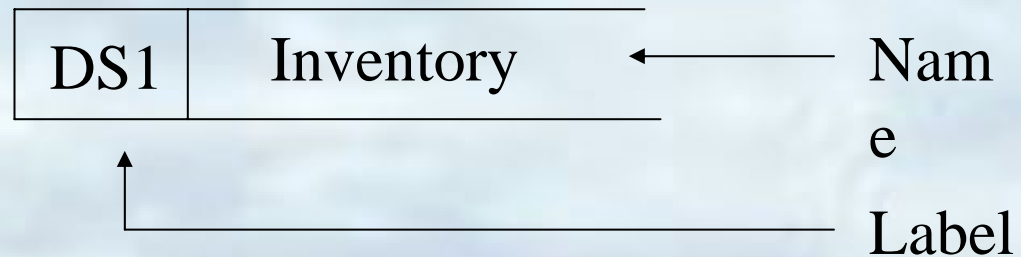
DATA FLOW DIAGRAMS

An alternate notation is often used

A Process



A Data store





GOOD STYLE IN DRAWING DFD

- Use meaningful names for data flows, processes and data stores.
- Use top down development starting from context diagram and successively leveling DFD
- Only previously stored data can be read
- A process can only transfer input to output. It cannot create new data
- Data stores cannot create new data

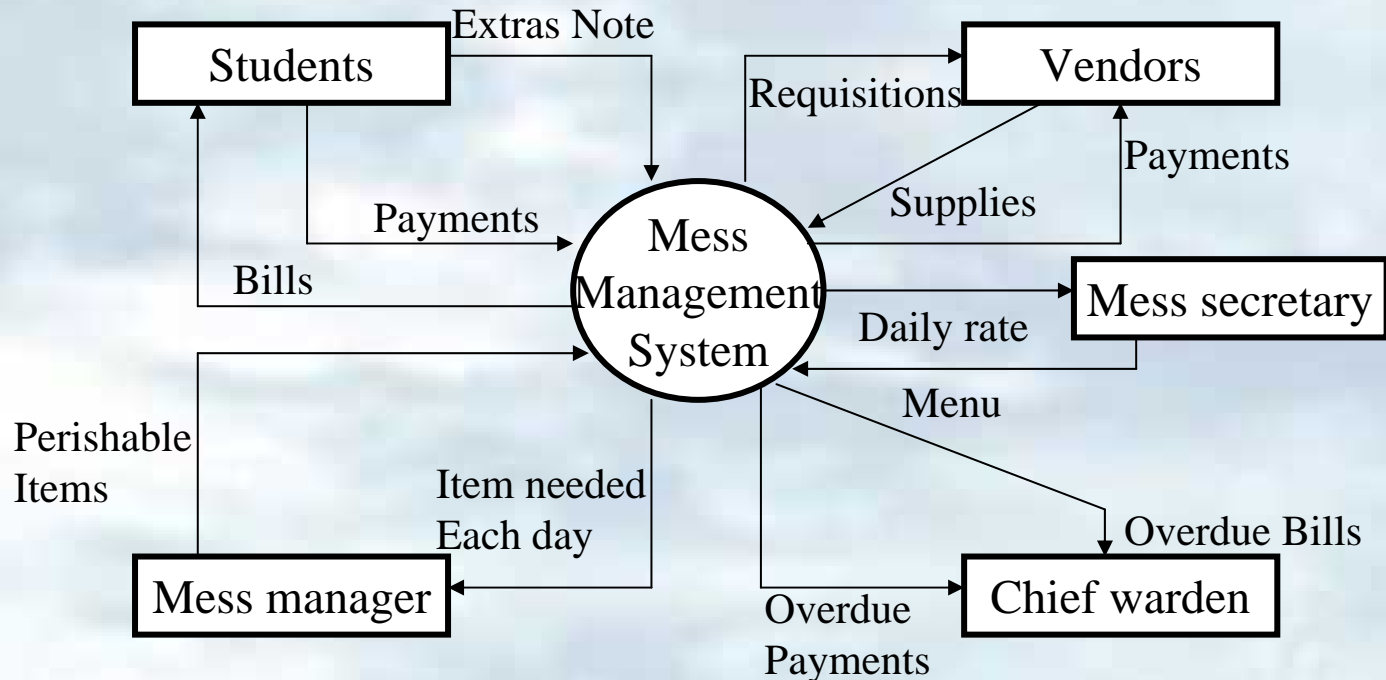


DESCRIBING A SYSTEM WITH A DFD

- An entire system is represented by one DFD which gives the system's overview
- It is called a context diagram
- It gives little detail & is also known as the top level DFD
- Context diagram of mess management is shown in the next transparency



CONTEXT DIAGRAM OF MESS MANAGEMENT SYSTEM



- Observe this diagram gives very little detail



LEVELLING DFD

- A context diagram gives an overview
- It should be split into major processes which give greater detail.
- Each major process is further split to give more detail.
- This process of giving more detail at a finer level is called levelling of DFD

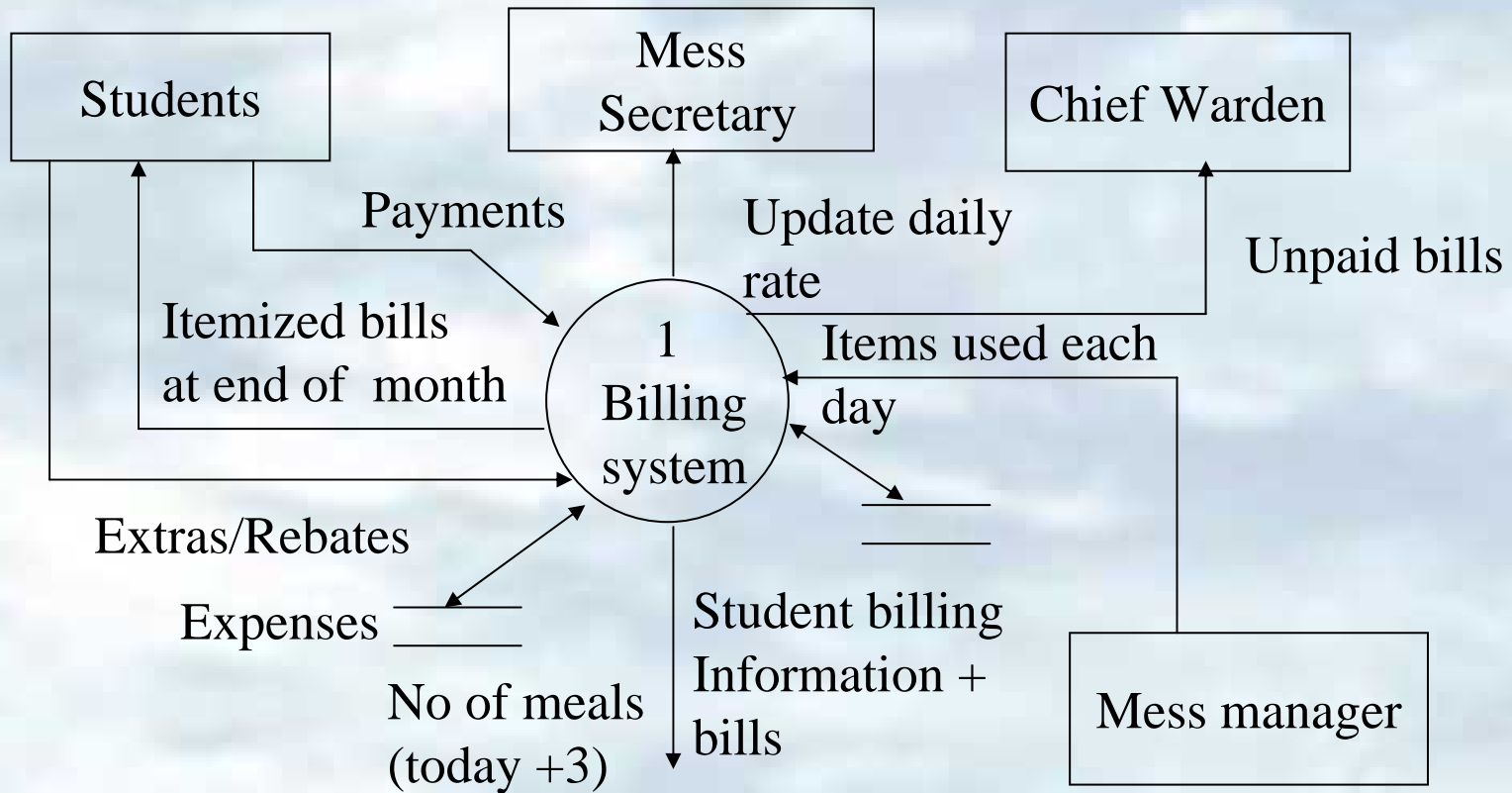


WHY LEVEL DFD?

- If a DFD is too detailed it will have too many data flows and will be large and difficult to understand
- Start from a broad overview. Expand to details - Idea similar to using procedures and linking these with a main program
- Each DFD must deal with one aspect of a big system



EXPANDED DFD FOR HOSTEL MESS MANAGEMENT



- Going to next process (Continued in next slide)

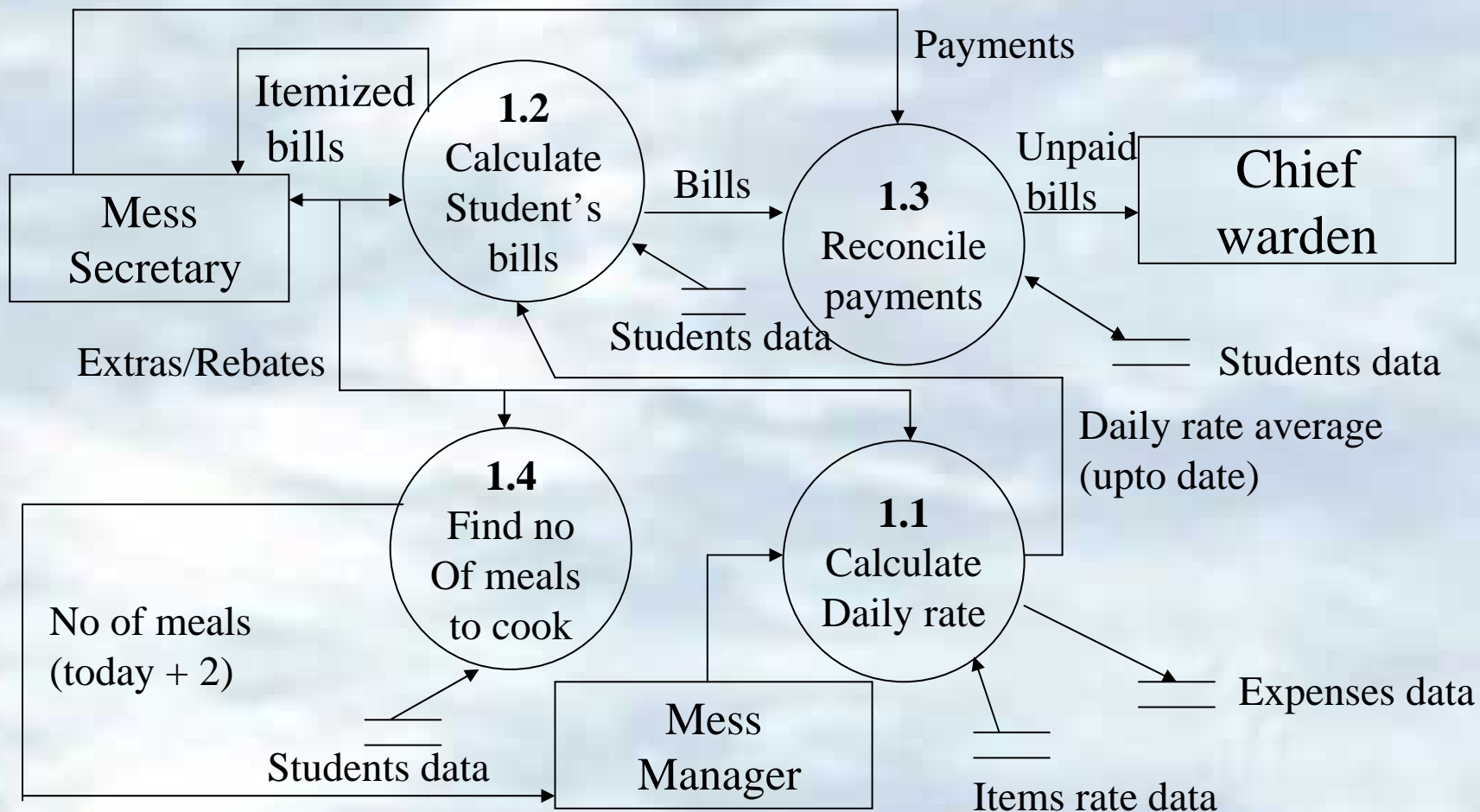
EXPANDED DFD FOR HOSTEL MESS MANAGEMENT

• **Continued** Low stock (today+2)





EXPANDED DFD-BILLING SYSTEM



- Observe numbering of processes



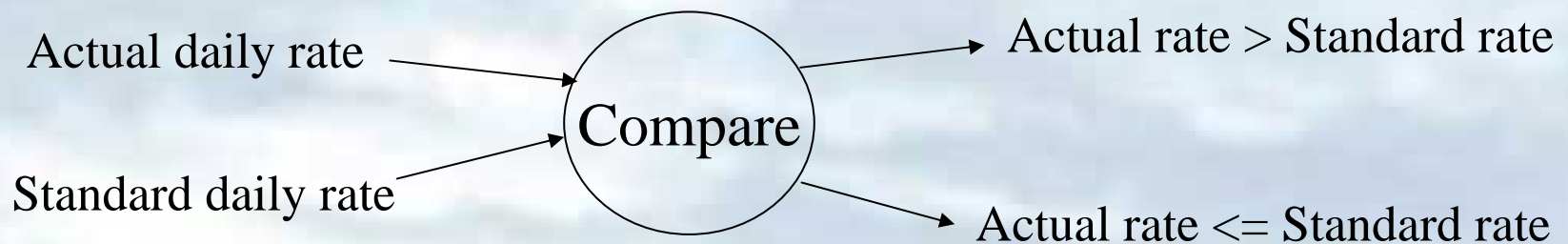
LEVELLING RULES

- If process p is expanded, the process at the next level are labeled as $p.1, p.2$ etc.
- All data flow entering or leaving p must also enter or leave its expanded version.
- Expanded DFD may have data stores
- No external entity can appear in expanded DFD
- Keep the number of processes at each level less than 7.



ILLEGAL CONSTRUCTS IN DFD

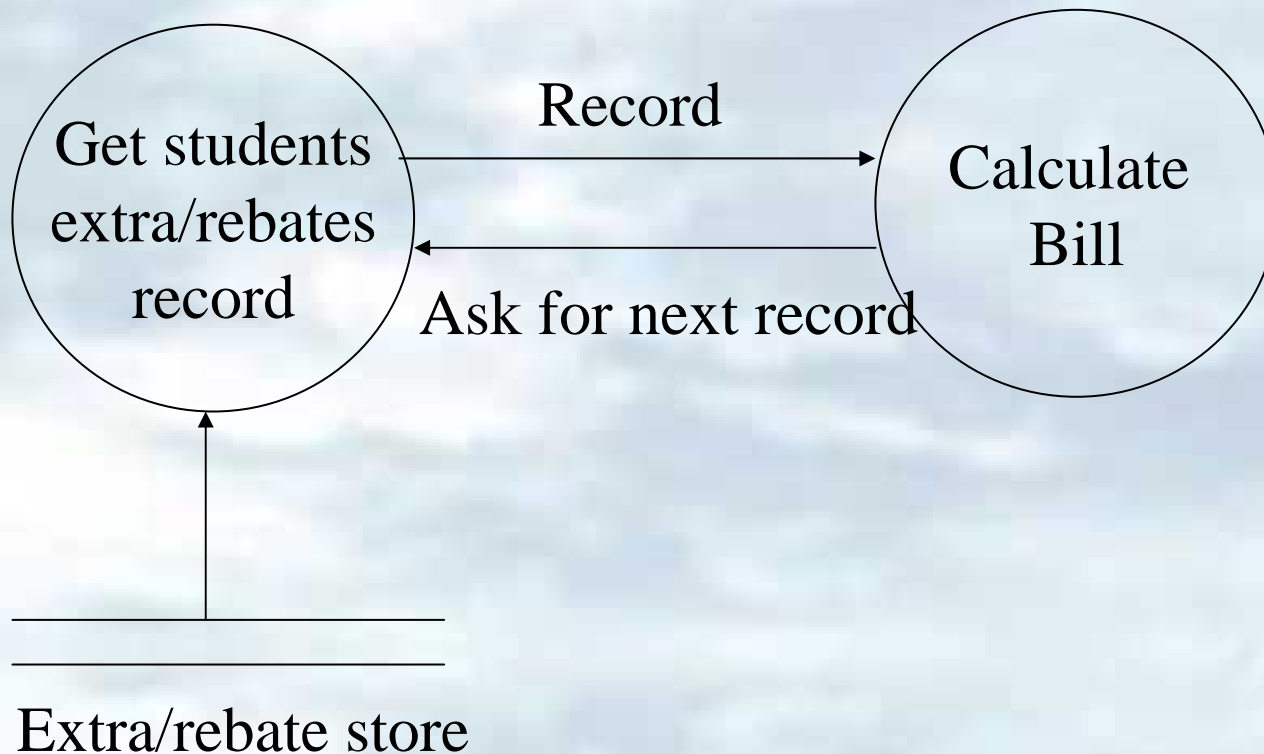
- No loops are allowed in DFD
- A process cannot be a pure decision



- A single data flow should not be split into many flows with different labels
- No data flow allowed between data stores



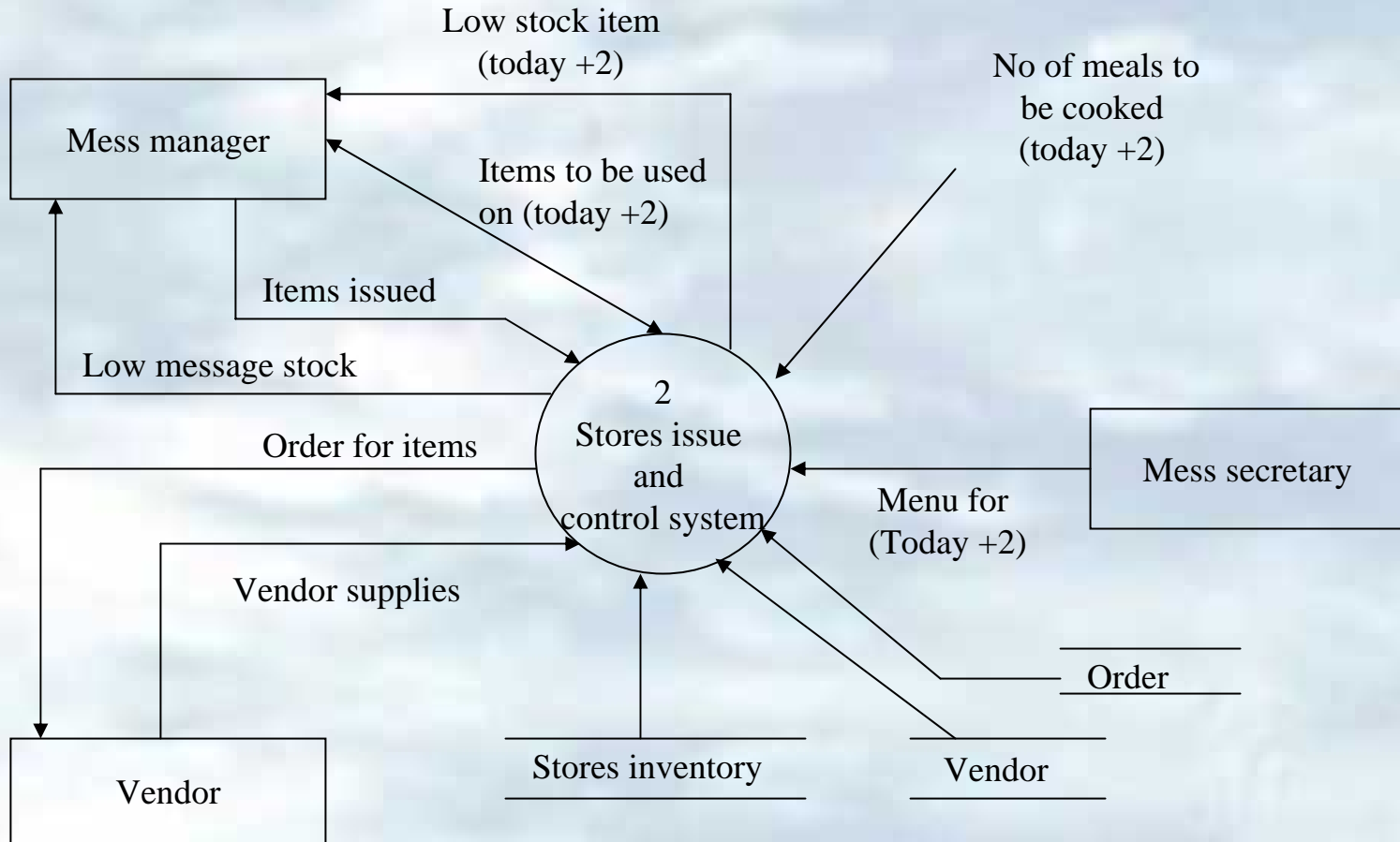
ILLEGAL CONSTRUCTS IN DFD



- Not correct as loop is formed



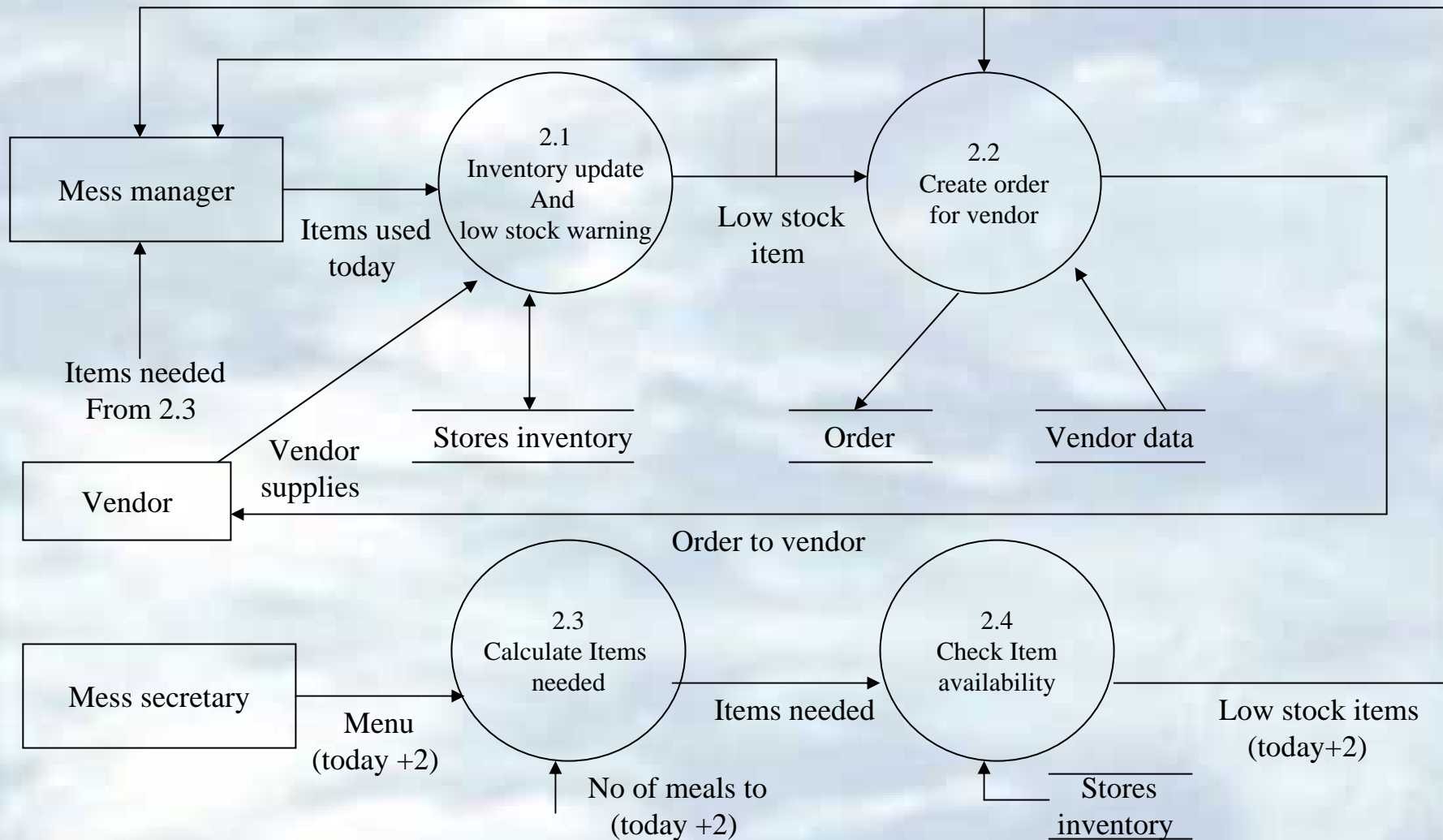
LEVELLING EXAMPLES



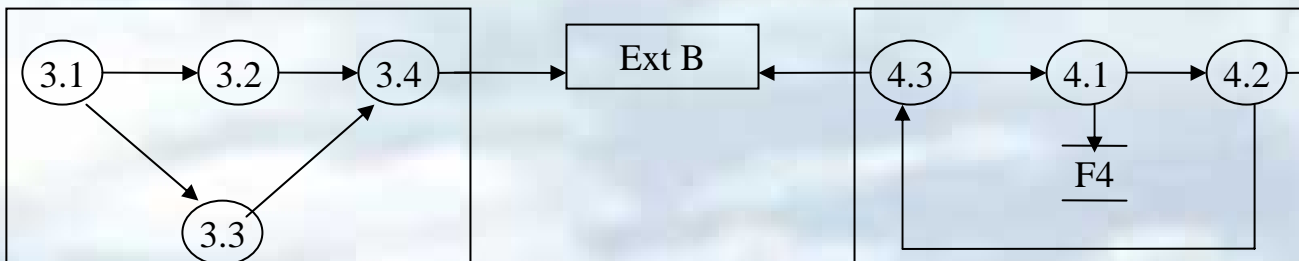
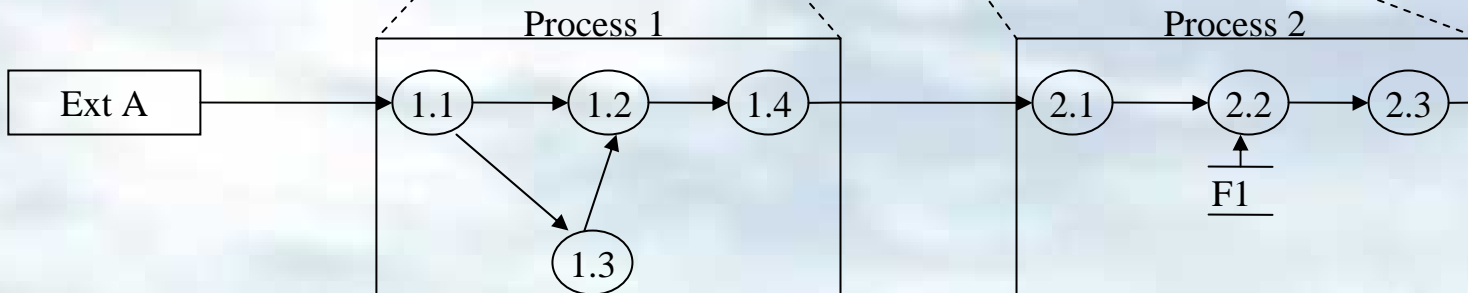
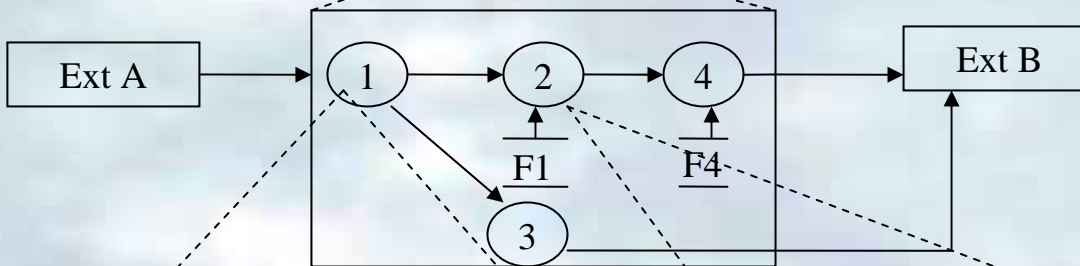
Stores issue control system process



LEVELLING EXAMPLES



LEVELLING EXAMPLES



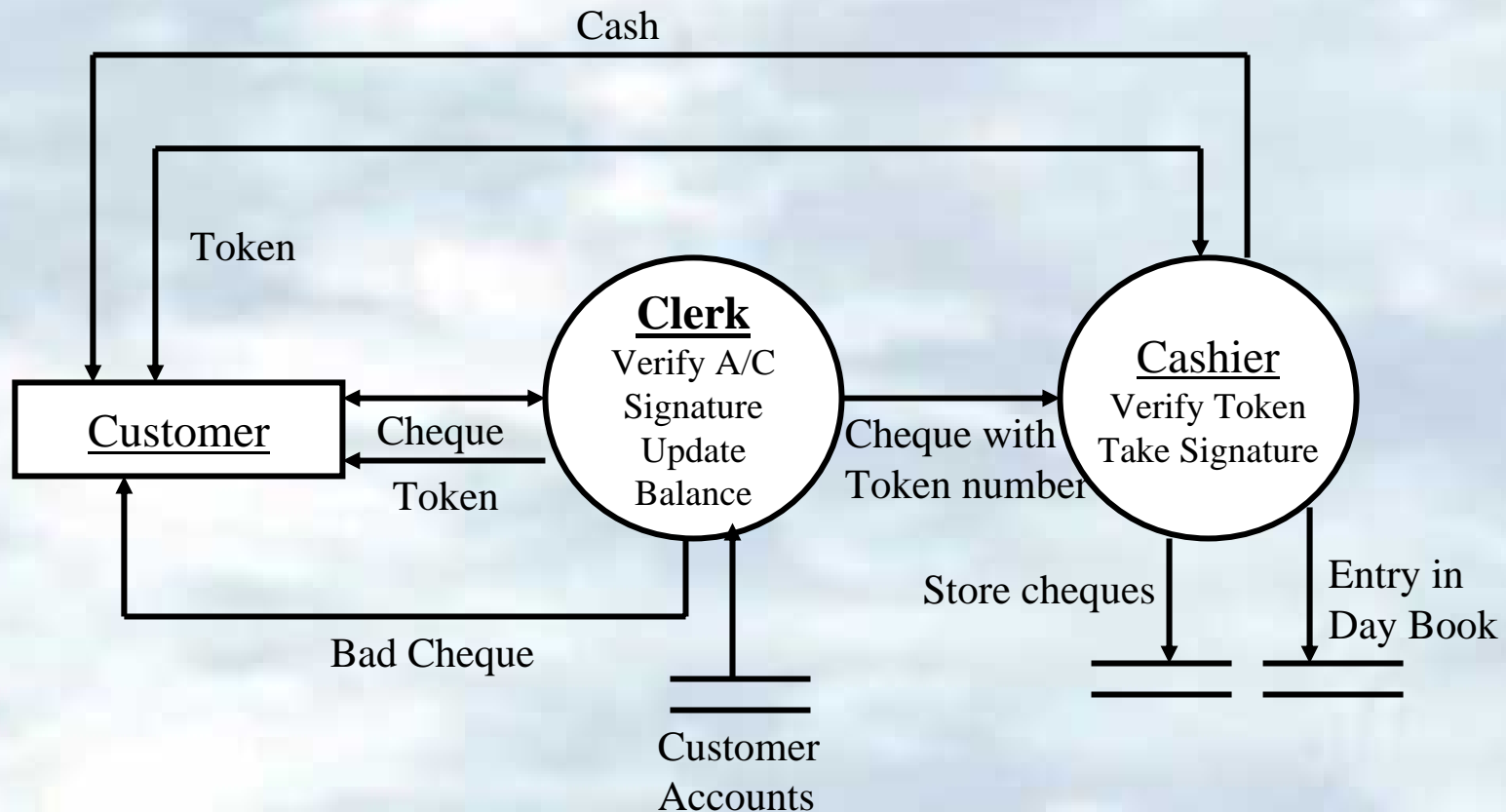


LOGICAL AND PHYSICAL DFD

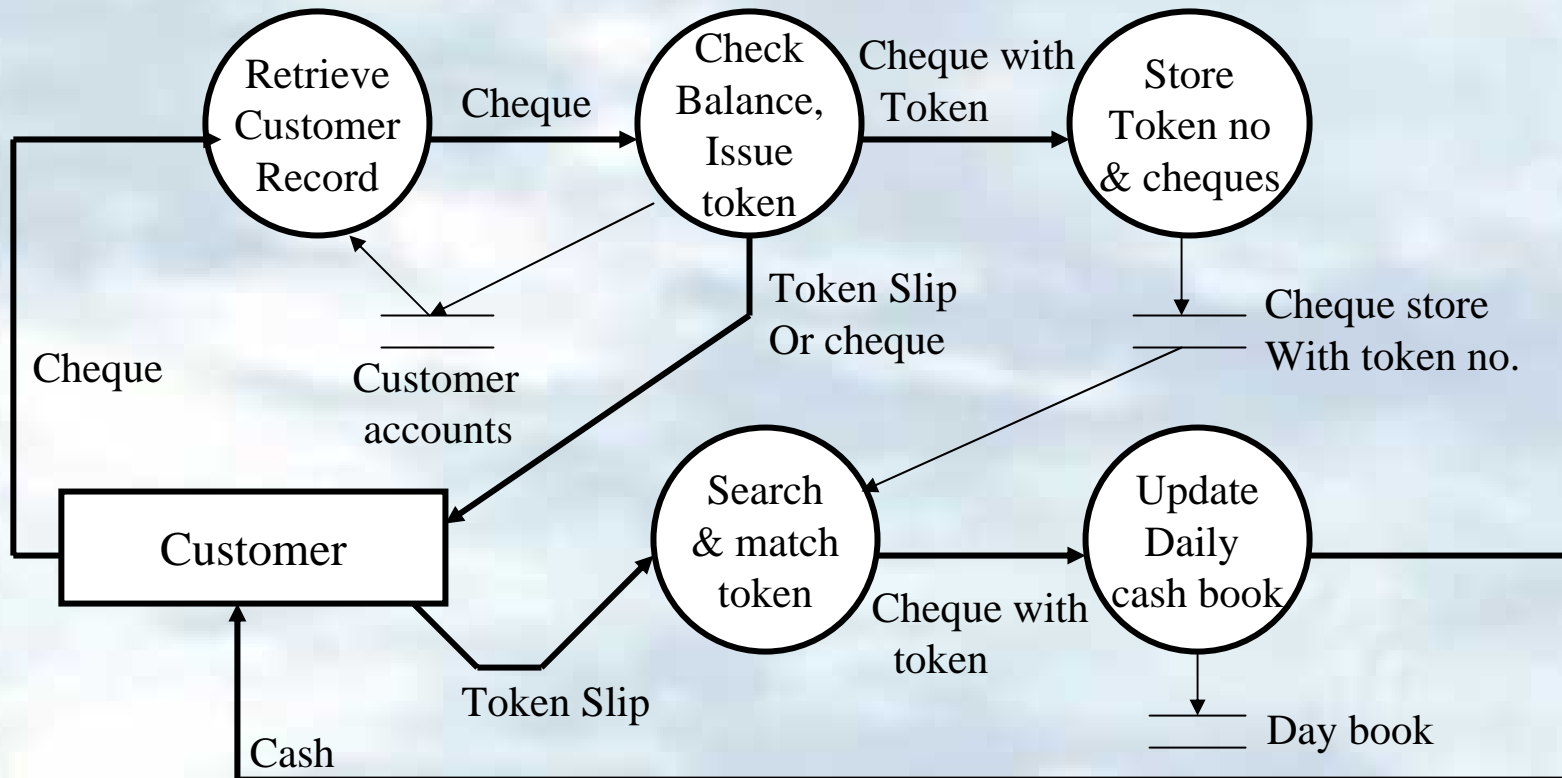
- DFD'S considered so far are called logical DFDs
- A physical DFD is similar to a document flow diagram.
- It specifies who does the operations specified by the logical DFD
- Physical DFD may depict physical movements of the goods
- Physical DFDs can be drawn during fact gathering phase of a life cycle



PHYSICAL DFD FOR ENCASHING CHEQUE

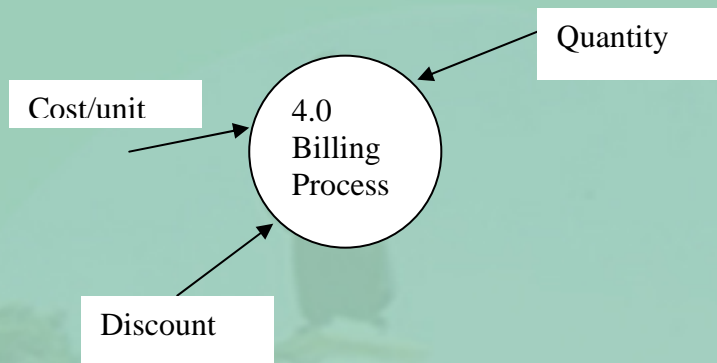


LOGICAL DFD FOR CHEQUE ENCASHMENT

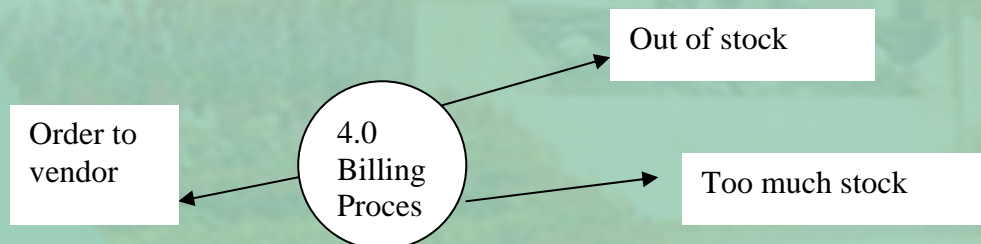


- 5.1 In a DFD external entities are represented by a**
- Rectangle
 - Ellipse
 - diamond shaped box
 - circle
- 5.2 A data flow can**
- only emanate from an external entity
 - only terminate in an external entity
 - may emanate and terminate in an external entity
 - may either emanate or terminate in an external entity but not both
- 5.3 A rectangle in a DFD represents**
- a process
 - a data store
 - an external entity
 - an input unit
- 5.4 External Entities may be a**
- source of input data only
 - source of input data or destination of results
 - destination of results only
 - repository of data
- 5.5 By an external entity we mean a**
- unit outside the system being designed which can be controlled by an analyst
 - unit outside the system whose behavior is independent of the system being designed
 - a unit external to the system being designed
 - a unit which is not part of a DFD
- 5.6 A data store in a DFD represents**
- a sequential file
 - a disk store
 - a repository of data
 - a random access memory
- 5.7 A data flow can**
- only enter a data store
 - only leave a data store
 - enter or leave a data store
 - either enter or leave a data store but not both
- 5.8 A data cannot flow between a store and**
- a store
 - a process
 - an external entity
- i and iii
 - i and ii
 - ii and iii
 - ii
- 5.9 Data cannot flow between two data stores because**
- it is not allowed in a DFD
 - a data store is a passive repository of data
 - data can get corrupted
 - they will get merged

- 5.10 Data cannot flow from an external entity to an external entity because**
- it will get corrupted
 - it is not allowed in DFD
 - an external entity has no mechanism to read or write
 - both are outside the context of the system
- 5.11 The following portion of a DFD is not correct as**
- there is no output data flow from the process
 - there are three data flow inputs to the process
 - there is no external entity
 - there is no data store



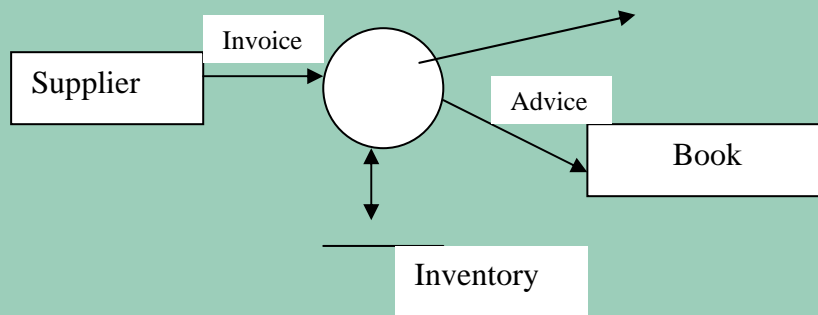
- 5.12 The following portion of a DFD is not correct as**
- there are many data flows out of the process
 - there are no input data flows to the process
 - the output does not go to an external entity
 - there is no data store



- 5.13 The following portion of DFD is wrong as**
- it has only one input
 - it writes and reads from the same data store
 - the process name is missing
 - output data flows to two external entities

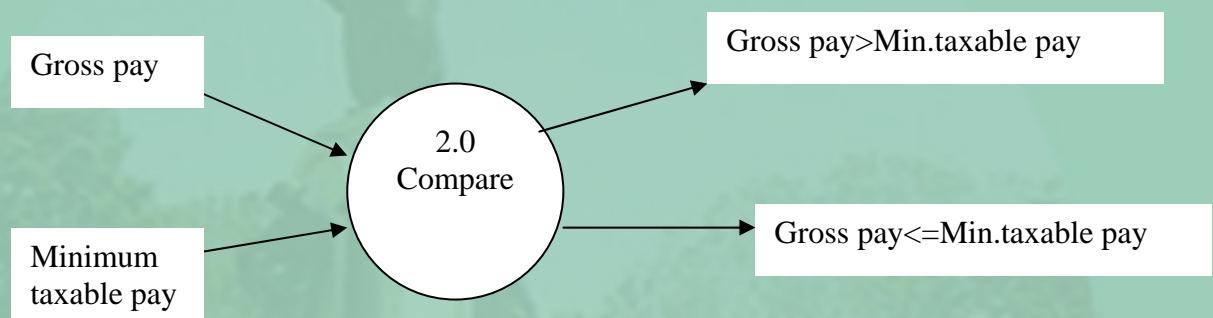
bill

Customer



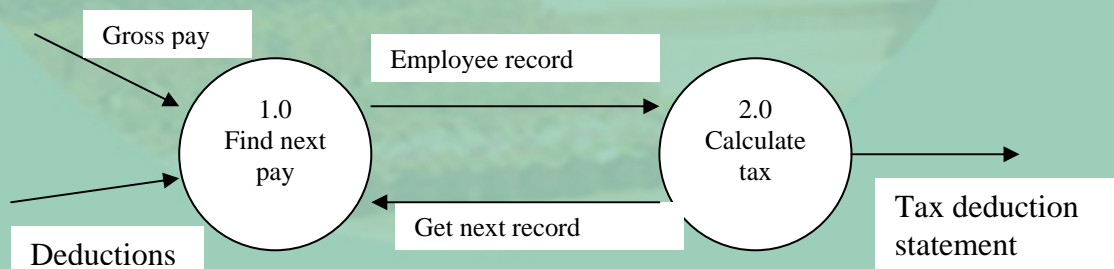
5.14 The following process diagram in a DFD is incorrect because

- the process is a single decision
- the process is not specified correctly
- there are too many input data flows
- the process does not refer to a data store



5.15 The following portion of a DFD is incorrect because

- the processes do not refer to a data store
- there is a loop between the two processes
- the processes are not specified correctly
- this structure is disallowed in a DFD



5.16 Data flow in a DFD must have

- an arrow showing direction of flow of data
- a meaningful name

- (iii) a label such as: xyz
 - (iv) no arrows as they are confusing
 - a. i and iii
 - b. ii and iv
 - c. iii and iv
 - d. i and ii
- 5.17 A context diagram**
- a. describes the context of a system
 - b. is a DFD which gives an overview of the system
 - c. is a detailed description of a system
 - d. is not used in drawing a detailed DFD
- 5.18 A context diagram is used**
- a. as the first step in developing a detailed DFD of a system
 - b. in systems analysis of very complex systems
 - c. as an aid to system design
 - d. as an aid to programmers
- 5.19 By levelling a DFD we mean**
- a. splitting it into different levels
 - b. make its structure uniform
 - c. expanding a process into one with more sub-processes giving more detail
 - d. summarizing a DFD to specify only the essentials
- 5.20 A DFD is normally levelled as**
- a. it is a good idea in design
 - b. it is recommended by many experts
 - c. it is easy to do it
 - d. it is easier to read and understand a number of smaller DFDs than one large DFD
- 5.21 A DFD is levelled by**
- a. examining complex processes in a DFD and expanding them into new DFDs with more processes which are easy to understand
 - b. merging a number of simple processes in a DFD into a complex processes in a new DFD
 - c. expanding the functions of a number of external entities into simpler functions
 - d. splitting a number of data flows into simpler data flows
- 5.22 When a DFD is levelled no new**
- a. data stores may appear
 - b. external entities may appear
 - c. processes may appear
 - d. data flows may appear
- 5.23 When a DFD is levelled**
- a. new external entities may be required
 - b. no new processes are allowed
 - c. no new data flows are allowed
 - d. new data stores may be necessary and are allowed
- 5.24 When a DFD is levelled it is a good idea not to**
- a. be concerned about the number of new processes at the next level
 - b. allow more than 5 to 10 new processes at the next level for each expanded process

- c. allow new data stores at the next level
 - d. allow any new processes at the next level
- 5.25 When a process is expanded during leveling**
- a. data flows entering it are replaced
 - b. all data stores used by it are replaced
 - c. all data flows entering it must also enter the levelled DFD
 - d. all external entities used by it are replaced
- 5.26 Before developing a logical DFD it is a good idea to**
- a. develop a physical DFD
 - b. develop a system flow chart
 - c. determine the contents of all data stores
 - d. find out user's preferences
- 5.27 A physical DFD specifies**
- a. what processes will be used
 - b. who generates data and who processes it
 - c. what each person in an organization does
 - d. which data will be generated
- 5.28 A physical DFD**
- a. has no means of showing material flow
 - b. does not concern itself with material flow
 - c. can show only stored material
 - d. can show the flow of material

Key to Objective Questions

5.1 a	5.2 c	5.3 c	5.4 b	5.5 c	5.6 c
5.7 c	5.8 a	5.9 d	5.10 d	5.11 a	5.12 b
5.13 c	5.14 a	5.15 b	5.16 d	5.21 b	5.22 a
5.23 c	5.24 d	5.25 a	5.26 b	5.27 d	5.28 b
5.29 c	5.31 a	5.32 b	5.33 d		



WORKED EXAMPLES

5.1 What is the main merit of DFD?

The main merit of DFD is that it provides an overview of what data flows in a system, what transformations are done on the data, what files are used and where results flow.

5.2 What is the role of DFD as a documentation aid?

It is a good documentation aid which is understood by both programmers and non-programmers (i.e., laypersons). As DFD specifies only what processes are performed and not how they are performed it is easily understood by a non-programming user.

5.3 What is a context diagram?

A diagram giving an entire system's data flows and processing with a single Process (circle) is called a context diagram.

5.4 What do you understand by levelling of DFD?

A context diagram is expanded into a number of inter-related processes. Each process may be further expanded into a set of inter-connected sub processes. This procedure of expanding a DFD is known as levelling.

5.5 What is a physical DFD?

A physical DFD specifies from where data flows and who processes the data and to whom the processed data is sent.

5.6 In what way is physical DFD useful?

It is easy to develop during fact gathering stage of systems analysis. Such a physical DFD is easily understood by a lay user who can verify the DFD drawn by an analyst and tell whether such a DFD corresponds to a particular operation taking place in an organization. Physical DFD is the starting point for developing the logical DFD.

5.7 What are the mistakes in each of the DFDs of Fig. 5.7(a)–5.7(e)? Correct these mistakes.

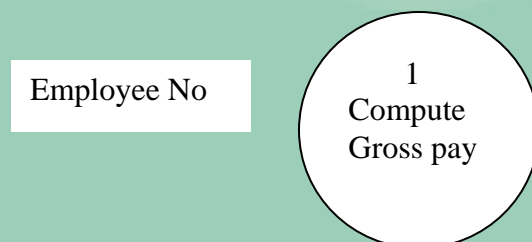


Fig 5.7(a)

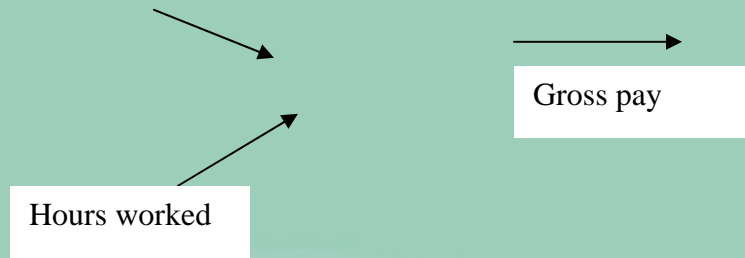


Fig 5.7(b)

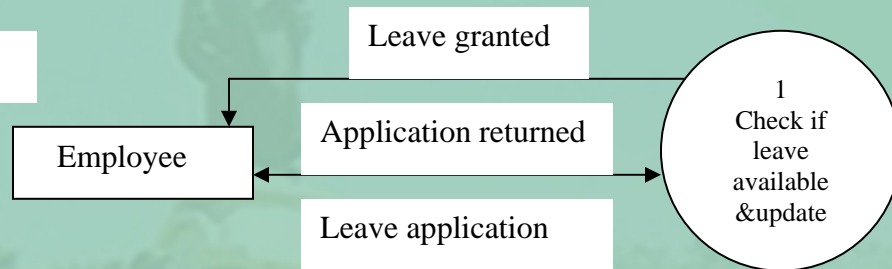


Fig 5.7(c)

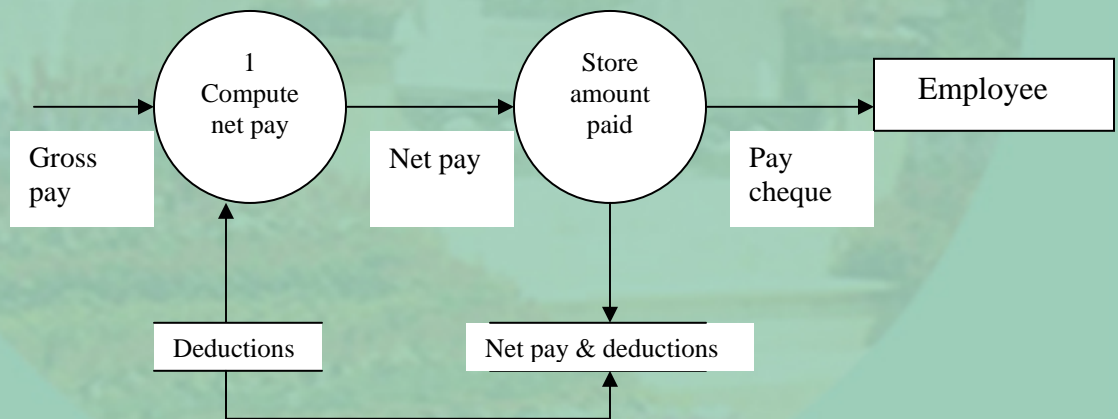
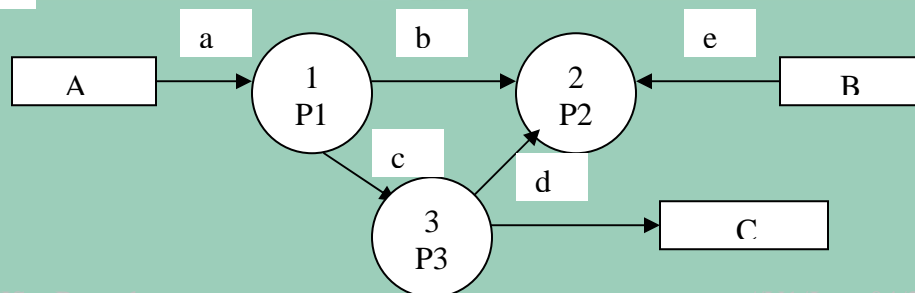
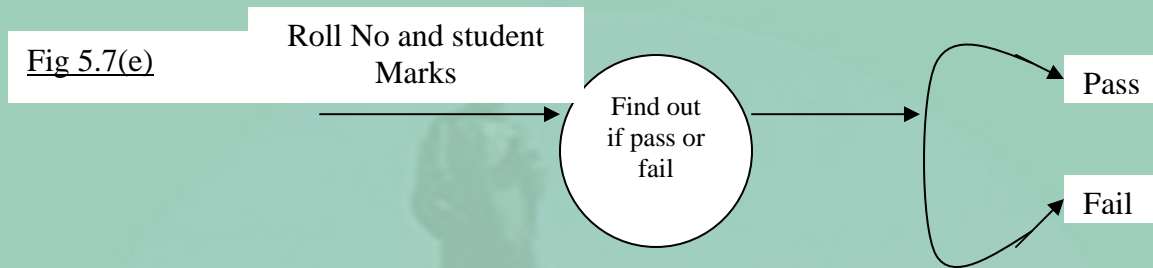


Fig 5.7(d)



f



- (i) To compute gross pay we need hours worked and hourly wage rate. This data flow is missing.
- (ii) A data flow cannot have two arrows pointing in opposite directions. A separate data flow line should be drawn for “application returned”.
- (iii) A data flow connects two distinct data stores without an intermediate processing step. (See also answer to Exercise 7.5)
- (iv) Process P2 has all input data flow and no output data flow.
- (v) The same data flow cannot be given two names.

- 5.1.1 In a DFD external entities are represented by a**
- (a) rectangle
 - (b) ellipse
 - (c) diamond shaped box
 - (d) circle
- 5.1.2 A data flow can**
- (a) only emanate from an external entity
 - (b) only terminate in an external entity
 - (c) may emanate and terminate in an external entity
 - (d) may either emanate or terminate in an external entity but not both
- 5.1.3 A rectangle in a DFD represents**
- (a) a process
 - (b) a data store
 - (c) an external entity
 - (d) an input unit
- 5.1.4 External Entities may be a**
- (a) source of input data only
 - (b) source of input data or destination of results
 - (c) destination of results only
 - (d) repository of data
- 5.1.5 By an external entity we mean a**
- (a) unit outside the system being designed which can be controlled by an analyst
 - (b) unit outside the system whose behavior is independent of the system being designed
 - (c) a unit external to the system being designed
 - (d) a unit which is not part of a DFD
- 5.1.6 A data store in a DFD represents**
- (a) a sequential file
 - (b) a disk store
 - (c) a repository of data
 - (d) a random access memory
- 5.1.7 A data flow can**
- (a) only enter a data store
 - (b) only leave a data store
 - (c) enter or leave a data store
 - (d) either enter or leave a data store but not both
- 5.1.8 A data cannot flow between a store and**
- (i) a store
 - (ii) a process
 - (iii) an external entity
- (a) i and iii
 - (b) i and ii
 - (c) ii and iii
 - (d) ii
- 5.1.9 Data cannot flow between two data stores because**
- (a) it is not allowed in a DFD
 - (b) a data store is a passive repository of data

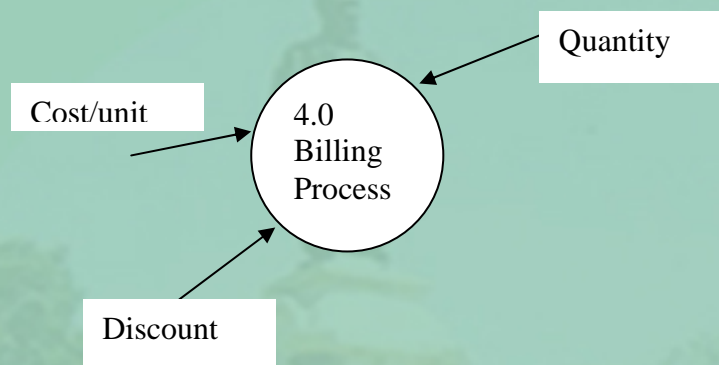
- (c) data can get corrupted
- (d) they will get merged

5.1.10 Data cannot flow from an external entity to an external entity because

- (a) it will get corrupted
- (b) it is not allowed in DFD
- (c) an external entity has no mechanism to read or write
- (d) both are outside the context of the system

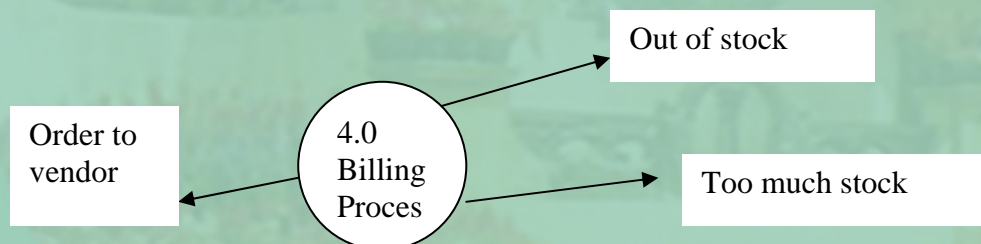
5.1.11 The following portion of a DFD is not correct as

- (a) there is no output data flow from the process
- (b) there are three data flow inputs to the process
- (c) there is no external entity
- (d) there is no data store



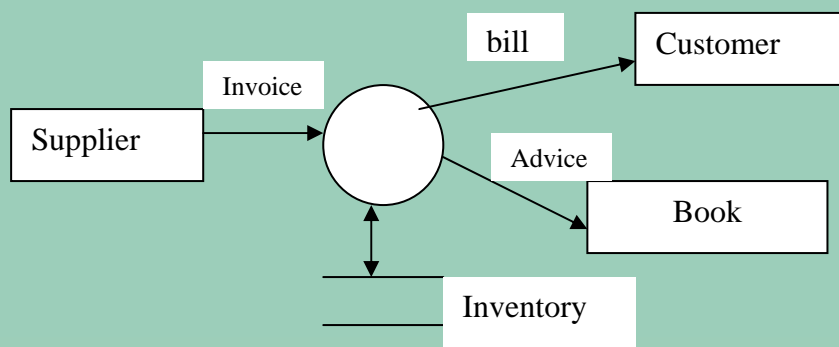
5.1.12 The following portion of a DFD is not correct as

- (a) there are many data flows out of the process
- (b) there are no input data flows to the process
- (c) the output does not go to an external entity
- (d) there is no data store



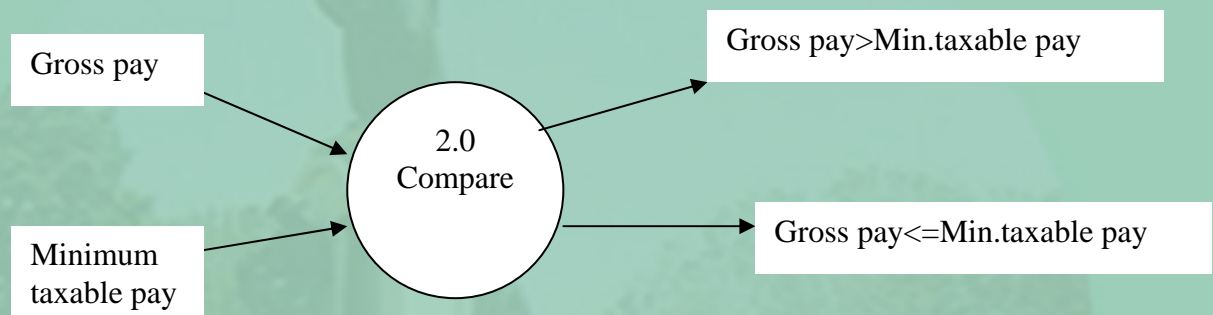
5.1.13 The following portion of DFD is wrong as

- (a) it has only one input
- (b) it writes and reads from the same data store
- (c) the process name is missing
- (d) output data flows to two external entities



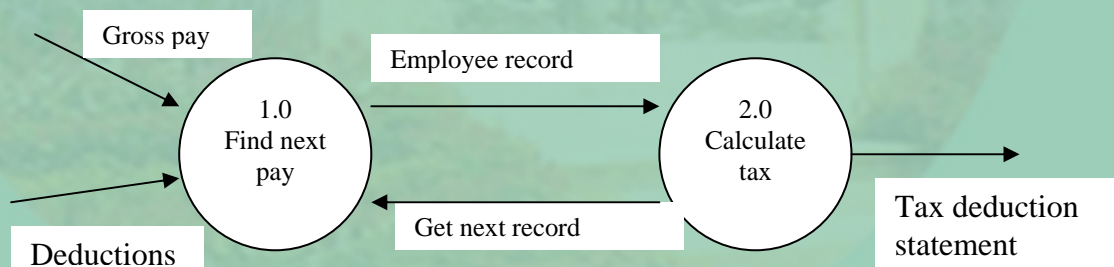
5.1.14 The following process diagram in a DFD is incorrect because

- (a) the process is a single decision
- (b) the process is not specified correctly
- (c) there are too many input data flows
- (d) the process does not refer to a data store



5.1.15 The following portion of a DFD is incorrect because

- (a) the processes do not refer to a data store
- (b) there is a loop between the two processes
- (c) the processes are not specified correctly
- (d) this structure is disallowed in a DFD



5.1.16 Data flow in a DFD must have

- (i) an arrow showing direction of flow of data
 - (ii) a meaningful name
 - (iii) a label such as: xyz
 - (iv) no arrows as they are confusing
- (a) i and iii
 - (b) ii and iv
 - (c) iii and iv

(d) i and ii

5.2.1 A context diagram

- (a) describes the context of a system
- (b) is a DFD which gives an overview of the system
- (c) is a detailed description of a system
- (d) is not used in drawing a detailed DFD

5.2.2 A context diagram is used

- (a) as the first step in developing a detailed DFD of a system
- (b) in systems analysis of very complex systems
- (c) as an aid to system design
- (d) as an aid to programmers

5.2.3 By levelling a DFD we mean

- (a) splitting it into different levels
- (b) make its structure uniform
- (c) expanding a process into one with more sub-processes giving more detail
- (d) summarizing a DFD to specify only the essentials

5.2.4 A DFD is normally levelled as

- (a) it is a good idea in design
- (b) it is recommended by many experts
- (c) it is easy to do it
- (d) it is easier to read and understand a number of smaller DFDs than one large DFD

5.2.5 A DFD is levelled by

- (a) examining complex processes in a DFD and expanding them into new DFDs with more processes which are easy to understand
- (b) merging a number of simple processes in a DFD into a complex processes in a new DFD
- (c) expanding the functions of a number of external entities into simpler functions
- (d) splitting a number of data flows into simpler data flows

5.2.6 When a DFD is levelled no new

- (a) data stores may appear
- (b) external entities may appear
- (c) processes may appear
- (d) data flows may appear

5.2.7 When a DFD is levelled

- (a) new external entities may be required
- (b) no new processes are allowed
- (c) no new data flows are allowed
- (d) new data stores may be necessary and are allowed

5.2.8 When a DFD is levelled it is a good idea not to

- (a) be concerned about the number of new processes at the next level
- (b) allow more than 5 to 10 new processes at the next level for each expanded process
- (c) allow new data stores at the next level
- (d) allow any new processes at the next level

5.2.9 When a process is expanded during levelling

- (a) data flows entering it are replaced
- (b) all data stores used by it are replaced

- (c) all data flows entering it must also enter the levelled DFD
- (d) all external entities used by it are replaced

5.3.1 Before developing a logical DFD it is a good idea to

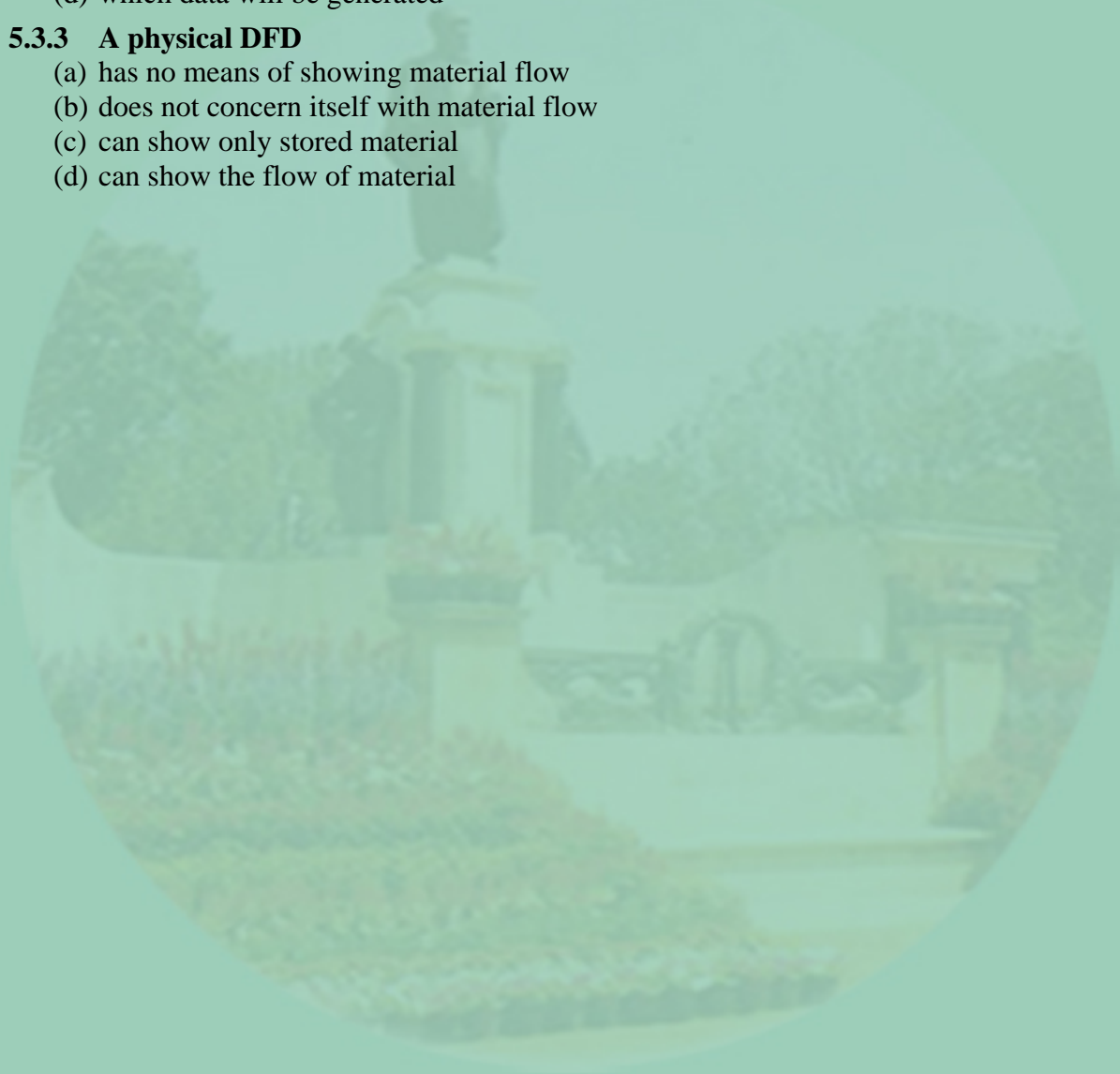
- a) develop a physical DFD
- b) develop a system flow chart
- c) determine the contents of all data stores
- d) find out user's preferences

5.3.2 A physical DFD specifies

- (a) what processes will be used
- (b) who generates data and who processes it
- (c) what each person in an organization does
- (d) which data will be generated

5.3.3 A physical DFD

- (a) has no means of showing material flow
- (b) does not concern itself with material flow
- (c) can show only stored material
- (d) can show the flow of material



KEY TO OBJECTIVE QUESTIONS

5.1.1	a	5.1.2	c	5.1.3	c	5.1.4	b	5.1.5	c	5.1.6	c
5.1.7	c	5.1.8	a	5.1.9	d	5.1.10	d	5.1.11	a	5.1.12	b
5.1.13	c	5.1.14	a	5.1.15	b	5.1.16	d	5.2.1	b	5.2.2	a
5.2.3	c	5.2.4	d	5.2.5	a	5.2.6	b	5.2.7	d	5.2.8	b
5.2.9	c	5.3.1	a	5.3.2	b	5.3.3	d				



QUESTION BANK

- 5.1 What is the difference between an external entity and a process in a DFD?
- 5.2 What are the main merits of using a DFD?
- 5.3 What is the role of DFD as a documentation aid?
- 5.4 What is the difference between a source and a sink in a DFD?
- 5.5 Why is data flow between data stores not allowed in DFD?
- 5.6 Is data flow allowed between an external entity and a data store. If your answer is "no", explain why.
- 5.7 What is a context diagram?
- 5.8 What do you understand by leveling of DFD?
- 5.9 What is the main difference between a flow chart and a DFD?
- 5.10 What do you understand by top down development of a DFD?
- 5.11 What is a physical DFD?
- 5.12 What is the difference between a physical and a logical DFD?
- 5.13 In what way is physical DFD useful?
- 5.14 Prepare physical and logical DFDs for the following activities:
- i. Issuing out a book from the library
 - ii. Returning a book to the library
 - iii. Getting a ticket reserved for a train journey
 - iv. Getting an item issued from a store
 - v. Getting your mark-sheet from a University office.
- 5.15 Admission procedure in a University is as follows:
- An advertisement is issued giving essential qualifications for the course, the last date for receipt of application, and the fee to be enclosed with the application. A clerk in the Registrar's office checks the received applications to see if mark-sheet and fee are

enclosed and sends valid applications to the concerned academic department. The department checks the application in detail and decides the applicants to be admitted, those to be put in the waiting list, and those rejected. Appropriate letters are sent to the Registrar's office which intimates the applicant. Give physical and logical DFDs corresponding to the above problem.

- 5.16 A magazine is published monthly and is sent by post to its subscribers. Two months before the expiry of subscription, a reminder is sent to the subscribers. If subscription is not received within a month, another reminder is sent. If renewal subscription is not received up to two weeks before the expiry of the subscription, the subscriber's name is removed from the mailing list and the subscriber informed. Obtain logical DFDs for this problem
- 5.17 Obtain a flowchart for Exercise 5.17 and state in what way it differs from the DFD.
- 5.18 Obtain a physical DFD for a simple payroll system described below. A list of employees with their basic pay is sent to a clerk. He calculates the gross pay using standard allowances which are known for each pay slab. Deduction statements such as loan repayment, subscription to association etc. are also sent to another clerk who matches these slips with the slips of gross pay and calculates net pay. This slip is used by a third clerk to write out pay cheques for each employee and sent to respective employees. The total pay bills paid are also computed.
- 5.19 If the procedure of Exercise 5.19 is to be computerised, obtain a logical DFD for the computer-based system.

Annotated References

1. Ian Sommerville, "Software Engineering", 5th Edition, Addison-Wesley, 1996, has a brief discussion of Data Flow Models on pp.101 to 103.
2. T.DeMarco, "Structured Analysis and System Specification", Yourdon Press, 1978. this book written by the original developer of DFD modeling is a well written book. It is a good reference book.
3. E.Yourdon, "Modern Structured Analysis", Prentice Hall of India, New Delhi, 1996. Chapter 9 (pp.139 to 187) is a good treatment of Data Flow Diagrams. All the topics covered in this module are discussed in this chapter.
4. Hoffer, J.A., George, J.F. and Valacich J.S., "Modern Systems Analysis and Design", 3rd Edition, Pearson Education Asia, New Delhi, 2002. Chapter 8 (pp.241 to 271) has a good treatment of DFDs with a running example of a quick service restaurant. Different types of DFDs and logical checking of DFDs are discussed well.
5. K.E.Kendall and J.E.Kendall, "Systems Analysis and Design", 5th Edition, Pearson Education Asia, New Delhi, 2003. Chapter 9 is devoted to Data Flow Diagrams (pp.241 to 285). Has a number of problems at the end of the chapter which are quite interesting.