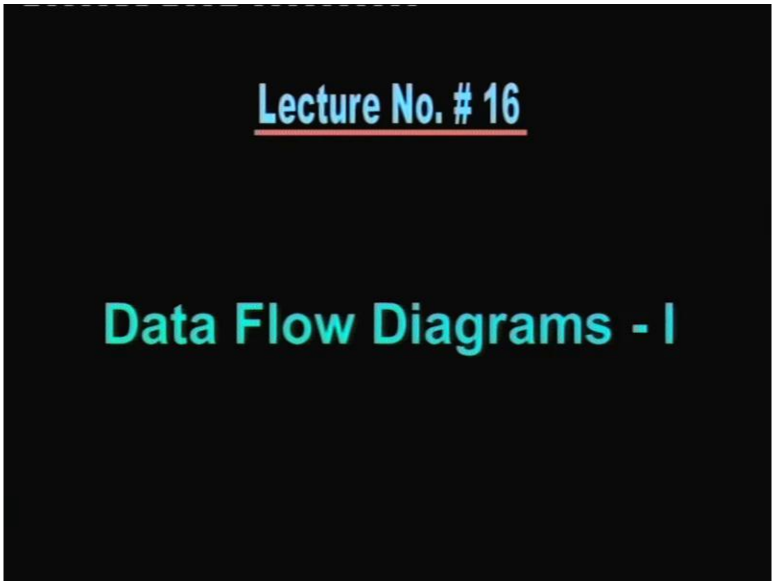
Management Information System
Prof. B. Mahanty
Department of Industrial Engineering & Management
Indian Institute of Technology, Kharagpur

Lecture - 16
Data Flow Diagrams - I

(Refer Slide Time: 00:51)



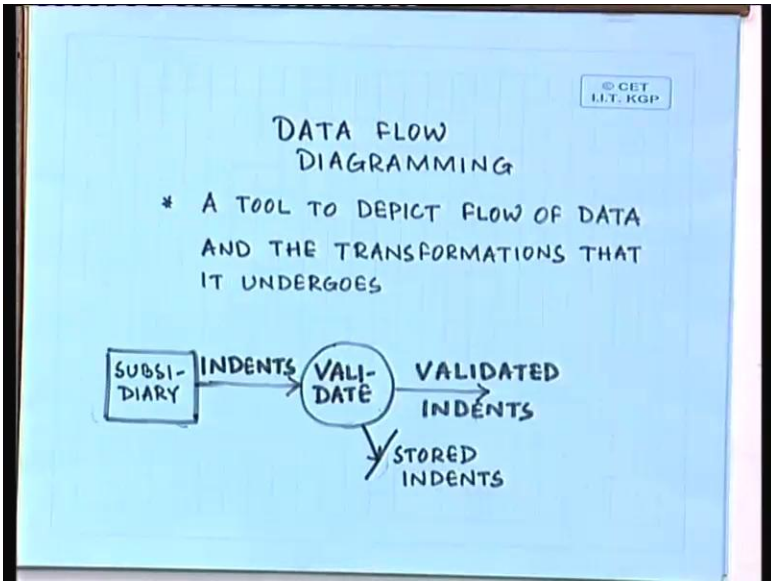
Lecture No. # 16

Data Flow Diagrams - I

Right, so today we are beginning a new topic that is data flow diagramming. Now in data flow diagramming what is happening? It is a tool by making use of which we can actually say, data flow diagramming. A tool to depict flow of data, what happens in the real life situation whenever we have let us say some sort of a system where we have you know various entities and between the entities the there is a flow of data, the diagram which depicts it is called a data flow diagram right.

So that is what I said that it is a tool to depict the flow of data and the transformations and the transformations that it undergoes. So there are two things one is a flow of data and another is the what transformation the data undergoes. Now see for example, let us see we have say we have a diagram like this where we have the indents and we have the validated indents.

(Refer Slide Time: 01:15)


© CET I.I.T. KGP

DATA FLOW DIAGRAMMING

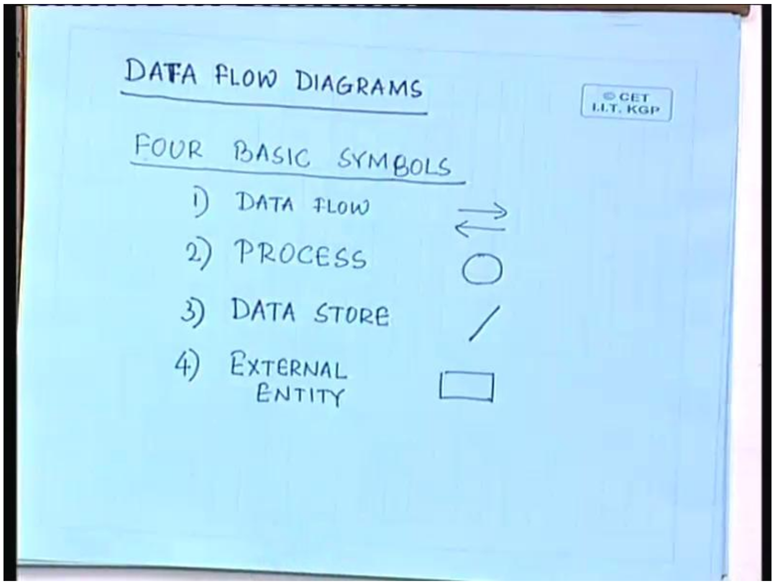
* A TOOL TO DEPICT FLOW OF DATA AND THE TRANSFORMATIONS THAT IT UNDERGOES

So if you look at these particular diagram you will see that the indents are you know going through a process of validation and we are having what is known as validated indents, all right. Fine, you go and get some pens from next room. So you can see that the data that is coming from now, we can do one more thing we can add here an entity. So these entity, let us say is the subsidiary, so what is happening we have the subsidiary, we have the indents and we have after validation, the validated indents.

So the entity is generating the data that is our indent these indent is coming to a process that is validate and the validation process is creating or transforming the indents to a new kind of a thing called validated indents all right. So the data flow diagramming is a tool which helps us in generating these kind of transformations. Now there is a third is a four thing in fact that four thing is let us say, when the indents are coming we are also storing them somewhere, so these are stored indents, these are stored indents.

So we can see here that we have some indents coming the validation process transforms the indents to validated indents and also it puts it in a store, right. So we can see therefore that the data flow diagram uses 4 basic symbols. It uses 4 basic symbols, the first one is the data flow symbol right the data flow symbol, the second one is called a process symbol that is shown in a circular manner, the third one is a data store symbol and the forth one is an external entity symbol.

(Refer Slide Time: 06:32)



DATA FLOW DIAGRAMS

© CET I.I.T. KGP

FOUR BASIC SYMBOLS

1) DATA FLOW ⇄
2) PROCESS ○
3) DATA STORE /
4) EXTERNAL ENTITY ▭

So you see just using 4 basic symbols that of data flow, data flow, process, data store and external entity. We can develop any amount of complex diagrams right. So this is the beauty of this particular technique that just using 4 basic symbols. Now before that let me give some basic idea that when we talk about data flow essentially it has to be a real flow of data rather than a control flow of data, what is the difference between a flow of data and a control flow of data or control flow, what is the difference control is, no no see data flow can be one directional, two directional or control flow also could be two directional there is nothing like that, I think you are thinking in a different way you have to think of a real system and think from that point of view.

Okay, let us look at it this way in the very first class if you recall I was talking about a physical flow and an information flow if you are recalling right and there I was telling that every physical flow or physical system has got an associated information system and in these particular lecture series, we shall consider or concentrate on the information side and not on the physical side.

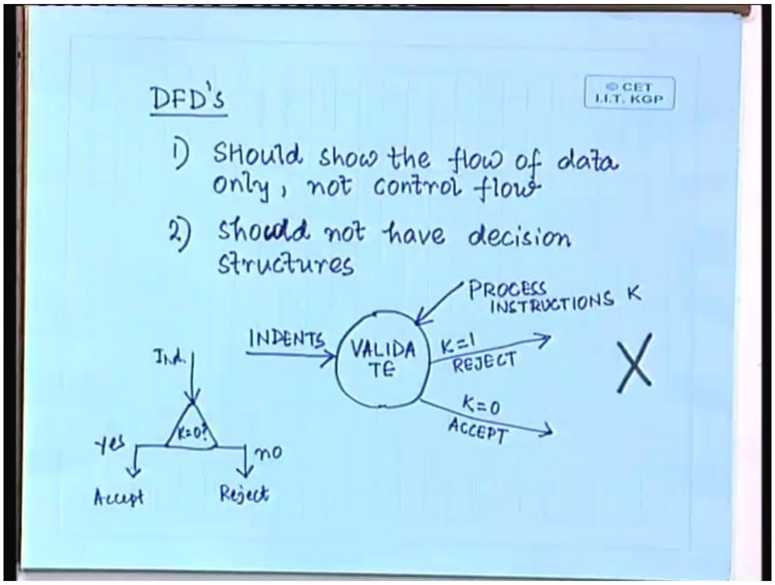
Let us say what is happening in the physical process. The physical process is that we have some indents coming, some material requests are coming all right and a purchase order is given and actual material is coming, you see when the actual material is coming along with that a goods received note, you see that you have received the actual material and you are generating a GRN or a goods received note and sending it to the appropriate places to indicate that material has come, all right.

So a particular document that goes along with it a GRN is what we are calling as data flow, if it is simply a request for GRN give me a some GRN, all right that is not a data flow all right or you give him some direction process information. Suppose you are the manager you are asking a clerk please process purchase order. So you can say a command instruction comes from the manager right we are not interested in those sort of things.

We are simply interested in the flow of documents what is really flowing right many a time, what happens particularly in programming you will see that when you are writing a program, you are sending from one program to another you know some kind of control logic that if may be in the form of a flag, see flag equal to yes, the program will act in one manner, flag equal to no program will act in another manner that is actually a control signal that you are sending, what you are sending is a control signal in the form of a flag but that is not a data, the data has to be a document it has to have a structure in the sense that it has some attributes, is it all right. So that is a data.

So in that sense the indent is a data that is flowing all right if you are returning the indent that is another kind of data that is flowing, simply saying that you know some sort of we are sending a flag if the flag is yes, then you process indent if the flag is no, then you do not process indent, no not that is not to be done. So some of the interesting things about data flow diagrams, let us write them here, in short we call what is known as DFD’s should show the flow of data only and not control flow this point already we have discussed. The second point should not have should not have decision structures, decision structures, see this is another thing many a time we are more interested in showing some sort of a decision structure. For example let us try to draw some decision structure see, look at this that indents process instructions K, K equal to 1 reject, K equal to 0 accept, all right.

(Refer Slide Time: 13:11)



DFD's

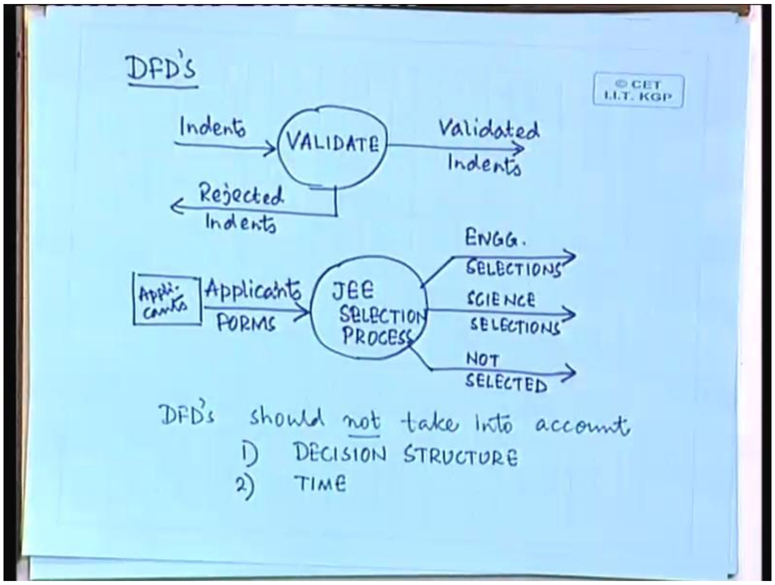
1) Should show the flow of data only, not control flow

2) should not have decision structures

So as if a branching kind of thing is taking place as if a bunching type of thing is taking place that I think you know what is a decision structure? A decision structure usually is like a triangle and then you have some kind of information coming here and yes, no all right. So if it is yes, you evaluate something K equal to 0 yes, no. As if these process is nothing but that know so we have our indents here and if yes, then we accept and no then we reject, look so we have the indent coming here then K equal to 0 yes, accept, no, reject. So this is a decision structure.

So these kind of now if you try to show these directly into a DFD and just put as process symbol here call it validate or something and then indents are coming and then process instruction and then, say K equal to 0, accept, K equal to 1, reject. This is actually showing the decision structure the idea is this is what is not to be done. So this is wrong it should not be done right, it should not be done. So what should be done if we do not do this then what we should do let us put that also. See, this is a valid data flow diagram, we have the indents, we have the validate process, we have the validated indents and we have the rejected indents. Actually, what is difference here is that the decision process is encircled or embedded all right into the process symbol.

(Refer Slide Time: 18:08)



DFD's

DFD's should not take into account
1) DECISION STRUCTURE
2) TIME

See that is the idea, now how these indents are validated, what is the process you know? What is a validated indent? What is not a validated indent? What is a rejected indent? It may so happen in a particular run all indents are validated none is rejected. So these rejected indents what does it mean does not mean anything but even then essentially, what we are writing is the inputs and all possible outputs of the process, is it all right. Say, suppose you want to show the another as an another example JEE selection process, applicants.

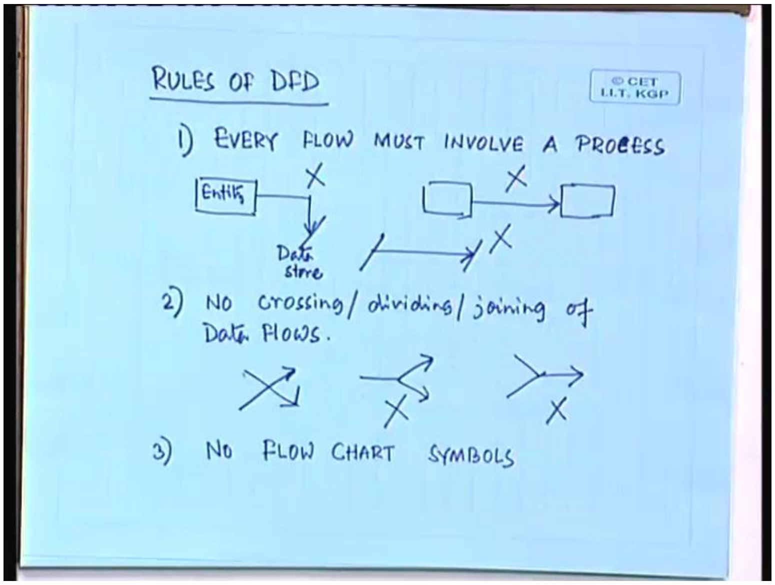
So I have just shown 3 categories engineering selection, science selection, not selected and you may go into hundreds of classifications, selected in mechanical engineering, selected in industrial engineering like that right. So you can you can keep on having many number of subdivisions whatever it may be but this JEE selection process, these applications really not applicants, you know applicants forms etcetera. Actually it should come from an entity called applicants. So you can see that this JEE selection process now, how the selection process bifurcates the applicants to various kinds of selections. See you can see that this is a global view.

So not only decision actually we should also mention that DFD’s should not take into account and number 2 is very important time, all right. So DFD’s should not take into account not only decision structure but also time, all right. So how time is coming that is you can see that engineering selections, science selections or not selected they are not actually happening in the same time as you are applying some of the people are not selected right in the beginning, they are not eligible, some are screened out in the screening test. So they are not selected at that time all right some are not selected in the main exams, all right some are not selected even after getting through IIT, I mean getting a rank but not getting a seat, fine.

So you see that not selected is not happening in one time it is happening at various times, fine. But if you want to make final distinctions you can always say further lines you can draw not selected in mains, not selected in screening, not selected in application stage itself etcetera, etcetera. So the basic idea is this process symbol only shows that how it is happening is actually hidden, all right. We are only showing what document is coming, what document is going out, fine. So this is how what we are basically achieving is the between the modules between the modules of the system, we are establishing that only there is a flow of data and nothing else not the decision structure not some sort of a command. The advantage that we achieve is they can actually be differentiated, they can be handled separately all right.

So one process can be separately treated then another process and you see these modularization is what we actually want this modularization is actually what we want at the end of the day. So that if we have to develop a large software, we can divide the job into smaller fragments and give it to various people to do. Then, let us see some of the advantages, some of the rules about the DFD’s which are important in the context. Rules are something like this, first of all every, every flow must involve a process. So this basically has a far reaching meaning what it means that if you are considering a flow that flow should be between a process to a process or a process to a data store or a process to an entity or entity to a process, data store to a process, is it clear, any other flow is illegal, should not be, clear what it means.

(Refer Slide Time: 26:05)



RULES OF DFD

© CET
I.I.T. KGP

1) EVERY FLOW MUST INVOLVE A PROCESS

Entity
Data store
X
X
X

2) No crossing/ dividing/ joining of Data Flows.

X
X

3) NO FLOW CHART SYMBOLS

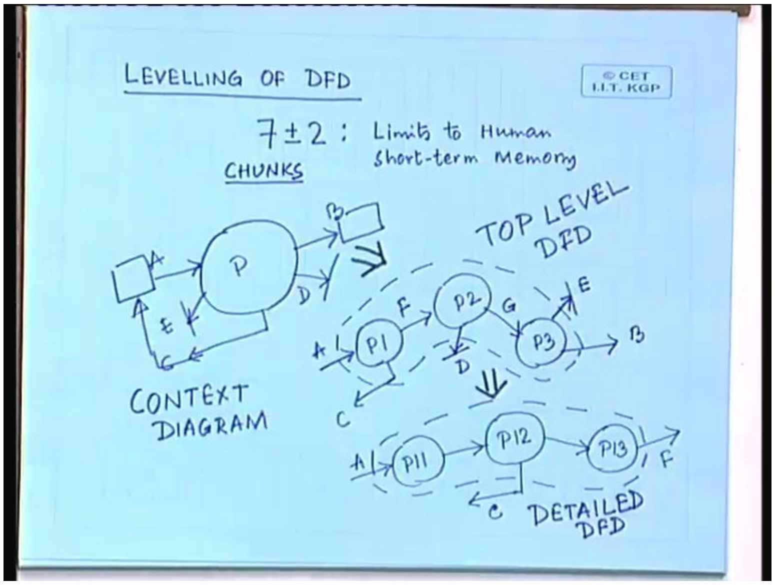
So you cannot show something like this but you may be tempted to do it, you may be tempted do this kind of wrong structures, anything else. So you see what is happening here, this is an entity, this is a data store. So if you show some data flow like from entity, let us say this entity is customer all right, think of an entity external entity like customer and think of a data store like customers information and assume the description, says the customer stores his information in a data file okay. So you are doing a purchase processing, now customer information is stored by customer himself, you are not doing anything all right.

So how can you put it to a process because a process depicts your activity, if you are not doing anything then why cannot you show like this? The difficulty here is that if some data processing is going on or information processing is going on which does not include the present system does not include the present system that has need not be shown that is not our concern, all right that is not our concern. What the data flow diagram depicts is only the processes, the flows and the data stores that are actually available in the current system and not outside the current system, is it okay so that is what we should do.

So in that sense all these are wrong cannot have entity to data store, entity to entity or data store to data store, data flow,, is it okay then second rule very important, no crossing, dividing or joining of data flows. So you cannot have something like you know this type or all these are wrong right. So you cannot, 2 data flows should not cross as such this is not a mistake, for say but this is a very wrong way of drawing a diagram, right. It is not a mistake is this crossing means you have placed your entities or things in such a manner that use just could not put them sufficiently properly, so that you could avoid the crossing of lines but with a little bit of thinking ye you can always avoid any, any open diagram you can always avoid the crossing of lines second is this one the dividing of lines is absolutely wrong, absolutely wrong if you have to a line is actually getting divided then there must be a process in between, all right. The process is doing something.

So that the line is actually getting bifurcated then the third one is two lines are joining again the same logic should not be happening then, finally no flow chart symbols. You have all drawn flow charts like I have shown those decision structures and others kind of things all these things should not be there at all, no flow chart symbol should be there, right.

(Refer Slide Time: 32:51)


LEVELLING OF DFD

7 ± 2 : Limits to Human Short-term Memory

CHUNKS

After this there is another concept which is known as leveling of DFD, actually the leveling of DFD is based on a concept of human decision making which is known as 7 plus minus 2, what is 7 plus minus 2 not 5 and 9 is 5 to 9, all right 5 to 9, all numbers beginning from 5 up to 9 is known as 7 plus minus 2 actually the 7 plus minus 2 is a very important number. We will discuss about it later while we shall discuss the humans as you know computing systems how, how humans perform in information processing because quite a bit, quite a bit of information is being processed by people themselves rather than by computer and moment people does it there are lots of errors and biases that comes in and these errors and biases are very important element of information systems, right because end of the day however automation computerization whatever you do you have to deal with people end of the day.

So this 7 plus minus 2is actually limits to these are limits to human short term memory. In computer language this is the RAM size of human brain, all right. This is a RAM size of human brain 5 to 9 very less in fact right now a days we talk about 512 MB RAM’s where 5 to 9 is nothing but please remember these 5 to 9 are not bits or bytes even these 7 plus minus 2 are actually expressed in terms of what is known as chunks. So human short term memory is expressed in terms of 7 plus minus 2, what is a chunk you see? A chunk could be just a bit it could be a byte it could be a single number or it could be a complete figure, all right it could be a complete figure entire page of a book something like this. The point is how you associate that particular thing with yourself all right.

So suppose a particular number is very familiar to you, suppose 4, 3, 9 what is this 4, 3, 9. Now suppose it means nothing to you then it is three chunks all right if 4, 3, 9 has no meaning to you then, it is 3 chunks but if 4, 3, 9 is highly significant to you, suppose it is some interesting or important number for you then it is only 1 chunk, is it all right. So that is why I said that this concept of chunk it could be a complete figure also all right. So this is how, so suppose I tell you a telephone number and I ask you go to the next room and tell it to the person sitting there all right you can remember because you are keeping it in your mind for few minutes or even few seconds but if I ask you this in the next class what is the number I told you will not remember because you have kept it in your short term memory, all right.

Now whatever that is another discussion but what is important in the even plus minus two as a concept for leveling of DFD because the short term memory size of human is so less to effectively deliver a information or communicate information, we should see to it that at a time we are not delivering more than 5 to 9 things to humans. So humans in other words, if you are drawing a diagram the diagram should not be having too many symbols essentially the data flow diagram is all about processes. So one diagram should show not more than 5 to 9 processes right if you do this then it will be easy for a person to comprehend is it all right see most of you if you have given an examination, you must know that you should not start a difficult problem from page one because if you start from page one, then you have to turn the page and you have to go to page two and if there are things which you have to keep referring then you have to keep turning the page you know you have to see it in page one again, go back to page two again, go back to page one which will not only you know increase the chance of making errors but also we will consume lot of time, all right.

So it is the same thing suppose you make a complicated diagram and then you have to understand this diagram, the discussions details you are reading then what will happen you have to keep going back to the diagram the diagram is too complicated. It is better instead of putting everything in one diagram, why not we put a number of diagrams series by series and explode the diagram you know in steps, this is known as leveling of DFD. So what is done here, suppose we have a process here, all right in this process actually having a transformation of A to B transformation of A to B, all right and also it generates C stores D somewhere in a file and also stores E.

So this is one complete view of the whole thing, all right. We have input A and output B transformation that takes place if you are rejecting it could be C, it could be indents validated indents and various stores which we example I have given. But in reality the process is little more complicated than this. You see what is happening here that A is coming A is getting transformed to P 1, P 1 is the process which transforms A to F and generates C then, F goes to a process P 2 where you store it to a data store D and generate G and in the process P 3, you are storing E and finally obtaining B, okay.

Now we can always try to put them also it will may not be a circle, see I have enclosed it if I enclose it then you see try to use your imagination, you will find that we are seeing this diagram. In another words, if we put a lens on P, you know powerful lens then we see lot of things happening inside, is it clear what I am trying to say. Fine, now similarly we can go one step further. You see now if A is the your indents, C is the rejected indents then what is happening here is as you receive A, the first process that is P 11 is the obtain the indents, classify, put them together, all right then P one to scrutinize, scrutinize, after scrutinize you reject C, then these are the scrutinized and then further validate, consolidate.

So you can see that likewise, we can always, we can always go from stage wise into more and more details. Now see this is the details of P 1 only similarly there will be details for P 2, there will be details for P 3. Now this is what is known as leveling of DFD that means seeing more and more detail as we are going stage by stage. The first one the top most level is usually called a context diagram, the second one is called top level, top level data flow diagram and the third one is called a detailed data flow diagram okay.

So the context diagram that is the first one the context diagram is actually it should show the entire system as one process. The beauty of the context diagram is that it should show the entire system as one process and I have not drawn here but it should also show the entities, it should also show the entities. So that means here C is actually going to this A only etcetera. So you see that context diagram should show the entities and from entities what information is flowing to the process, the overall process and what transformation the information is going through and also what are that various data stores which are used in the process, clear. So that is known as a context diagram the top level data flow diagram on the other hand is a little more detailed, little more detailed where you are showing individual processes and how things are happening there. Then, we have the detailed data flow diagram where the individual processes are further leveled. This process is known as leveling of DFD.

So when we take up examples, we shall start seeing how to draw the context diagram, how to draw a top level data flow diagram and how to draw detailed data flow diagram. Now the question is where to stop? You can again have P 11 and you can start drawing further detailed data flow diagram, like that you know you can keep on expanding the diagram to very very minute test levels. The idea is the smallest process should be such which is reasonably, what you say that easily can be translated to a single program that is the level where we should stop, where we would stop if P 11 is sufficiently small, so as to be represented by a single program right, what should be a single program approximately 50 lines of code, approximately 50 lines of code.

So if we can get down of to that level which can be implemented by a single program we should not expand it further, is it all right. So that is the rule where we should stop. Fine, so we stop here today what we have covered today is the basic ideas of data flow diagram and in the next class, we shall take up some examples. Thank you.