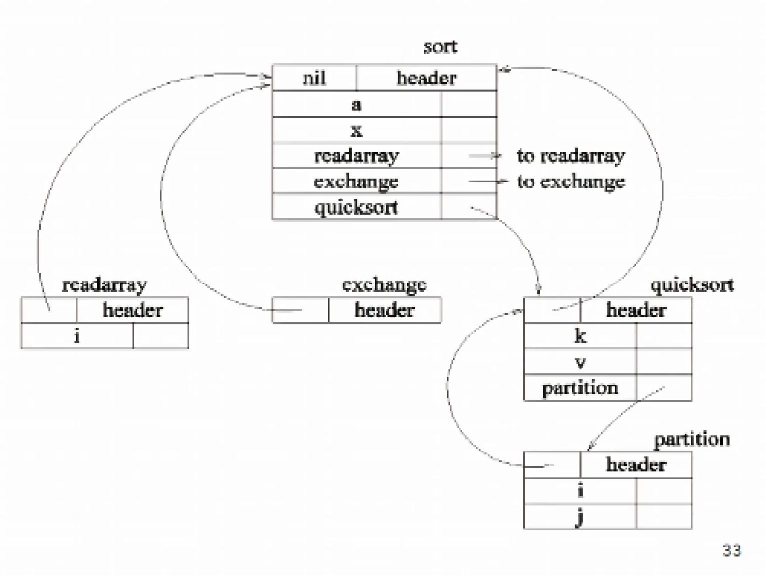
Indian Institute of Technology

Kanpur
NPTEL

National Programme
On
Technology Enhanced Learning
Scopes Title
Compiler Design
Lecture – 22

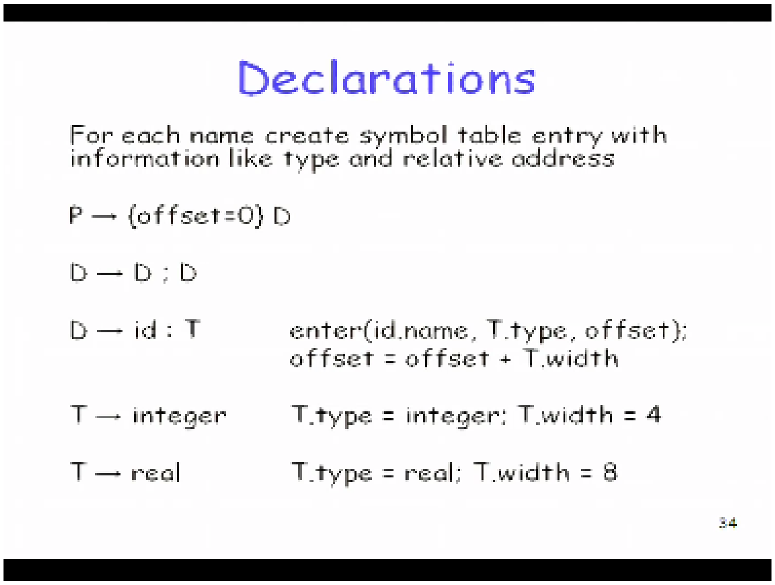
by…
Prof. S. K. Aggarwal
Dept. of Computer Science and Engineering.

(Refer Slide Time: 00:16)



So let us continue discussion on where we left from the previous class. Started looking at and look at tools of at least one single symbol table let us go for some of the scope this is how simple memories so we have single table first scope is at a highest level this is a programme okay I come to the programme so this is one of the scopes and then whenever I have declaration of function when I just make this entry and this is me pointer to the scope which is nested inside this and then again I can then I also have and let me look at the decorations. So this is how symbol is possible.

(Refer Slide Time: 01:20)



Declarations

For each name create symbol table entry with information like type and relative address

P → {offset=0} D	
D → D ; D	

D → id : T	enter(id.name, T.type, offset);
	offset = offset + T.width

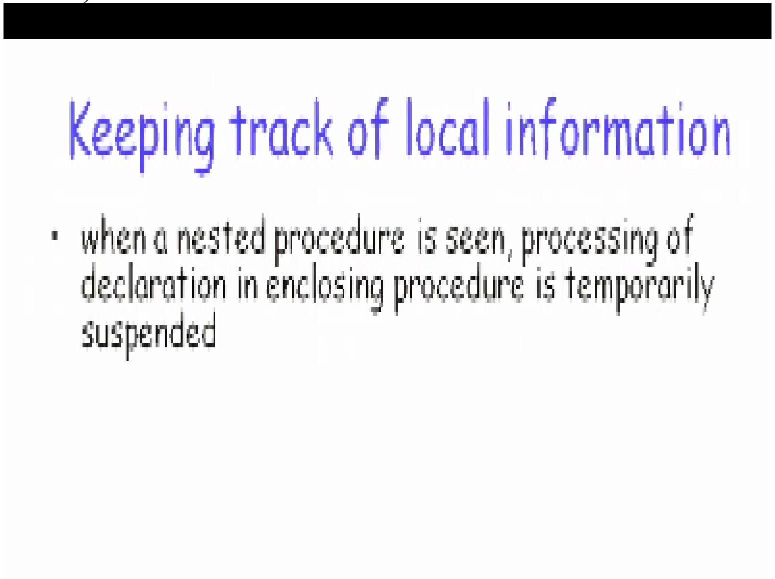
T → integer	T.type = integer; T.width = 4
T → real	T.type = real; T.width = 8

34

And then we looked at declarations and then we said that set that to begin with we will start with the top step with zero we are trying to get rid relocated reports and this phase that in each scope my offset is zero to begin with and how do I change this variable global variable every time. I sign some variable, I just add this value the offset and this ensures that whenever I have in the single table so for the first variable I am going to entry of set 0 here but suppose this variable will takes four bytes .

And the next variable will be entry a offset of space value right discussing and then we started looking at little more complex things so for arrays and so on now.

(Refer Slide Time: 02:16)

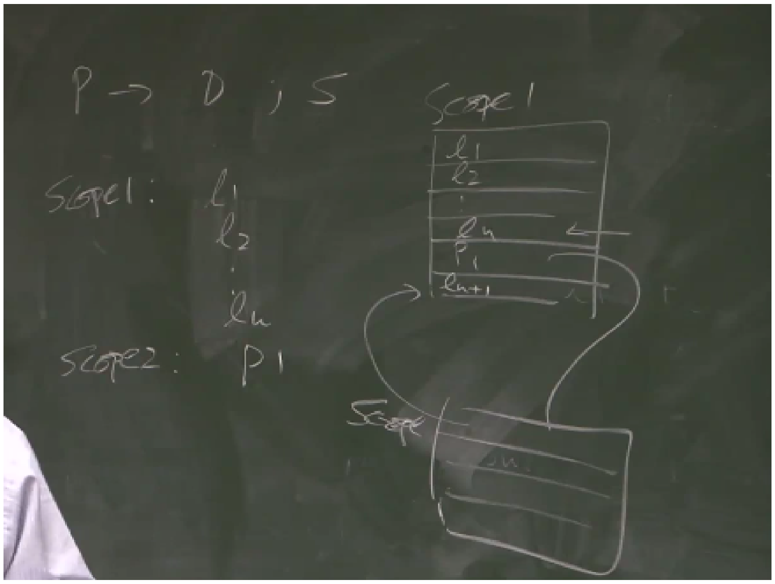


Keeping track of local information

- when a nested procedure is seen, processing of declaration in enclosing procedure is temporarily suspended

Now what we want to do is that means now it is possible that in addition to adding the program.

(Refer Slide Time: 02:25)



Which has some declarations followed by a statement I can also say that a procedure is nothing but a declaration at this point of time I am not interesting the body of the procedure we tell you that if I thought to see what is rapport which is going to be executed but if I just okay I had this also a procedure declaration then what will happen that because of this nesting so suppose what can happen is that I may have say let say scope one and here and have some string of local variables and once a string of local variables.

Then I may have procedure P1 so as soon as I have a procedure P1 what I need to do I have a to create a symbol T. So I can this am going to teach as a different scope okay, so this scope is inside scope one so what will happen is that I have this scope one declaration corresponding to L1L2 LN and then I have this declaration of T 1 which gives me a pointer to the simple table off scope 2 and then it should have its own local variables now first thing that you must realize is that when I start with a variable like offset with L1 with an offset of zero.

I will process it up to this point and as soon as I encounter T 1 and it has its own local variables want to be the offset I have to set it back with you because within this scope. I am again generating variables we are starting with an offset of zero but when I come back here now after this if I come back here and suppose. I have available and l n +1 then X offset must be computed with respect to the offset of L n so I must keep track of this information of offset of each of the scope.

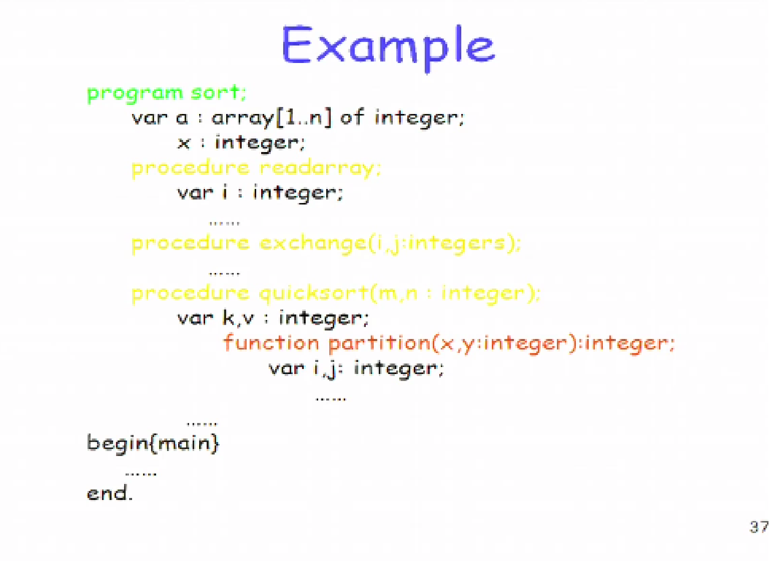
Okay as we mentioned previous class that statics perhaps is the data structure what I can do is I can leave now a state on which I keep offset on the top and when I reach this particular scope which gives me a new symbol table. I push now one more offset on top of whatever my state was and say that this is not offset of scope P1 which will be 0 again, I will keep on implementing and when I come back then I will pop this value and I will have the offset of L n on top and then I can start processing from there.

Now two kind of scenarios where I say that in this I must have all the declarations together and then all the procedure declarations but there some languages which say that I can have declaration and procedure declaration of maintained so I am general scenario where I can take care of both these , so this first problem definition here what we are trying to do before ,I start getting into the implementation and the course.

Yes no it is problem definition is here not sure first you have the procedure when you have a two classes so normally if you look at object oriental language this problem language here what will happen here that you will have a class definition and then you will have a another class definition either you have a situation , where I have a class definition and start it will finish and go on that would different so I am not pointing this point this time I am only that become later.

So we creating and we mixing several things first we are trying to simple table I am only pushing information about the offset on the static symbol table is separate right so I am saying that if I have declaration a sequence of declaration that I start with some scope and then I suppose to have let me give you a static example right away let take this program okay so what have here I am starting with a program sort here to the main program.

(Refer Slide Time: 08:17)



Example

```
program sort;
    var a : array[1..n] of integer;
        x : integer;
    procedure readarray;
        var i : integer;
           ......
    procedure exchange(i,j:integers);
           ......
    procedure quicksort(m,n : integer);
        var k,v : integer;
            function partition(x,y:integer):integer;
                var i,j: integer;
                    ......
         ......
begin{main}
     ......
end.
```

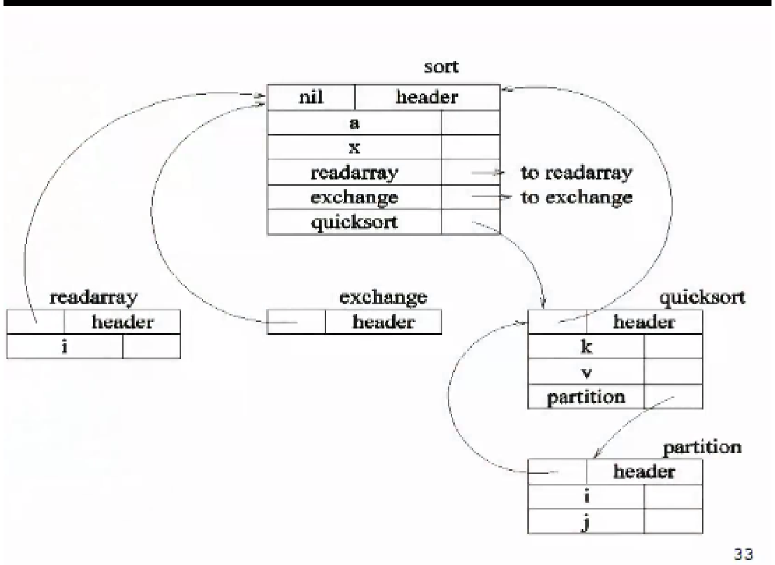
37

There are some variables which are local to solve and a with it sort I have nesting of in the string of procedure start reorder which as local variable and I told you a coding permission this point I am not interesting within that I have a procedure exchange which takes two arguments that I and J are you but has no local variable and then I have another procedure quick sort which has two arguments and which also has it is local variable and within this I have function partition at this point of time is I am only trying to create a symbol table such that at the top. I have a symbol table corresponding to sort which has variables A and X and which also has variables redirect exchange and quick sort.

For these three variables point to symbol tables three different symbols and then from these symbol tables, I also have a pointer right so I am keeping all the symbol tables are not throwing anything what I am trying to remember here is that when I say that A is the first variable I am pumped up and therefore at a locate certain space for this is going to be at an offset of zero because that is the first variable and then I will compute how many bytes this will take so if I say that this is going to take me here takes four bytes and this going to take 4 and therefore X must be in the offset of 4.

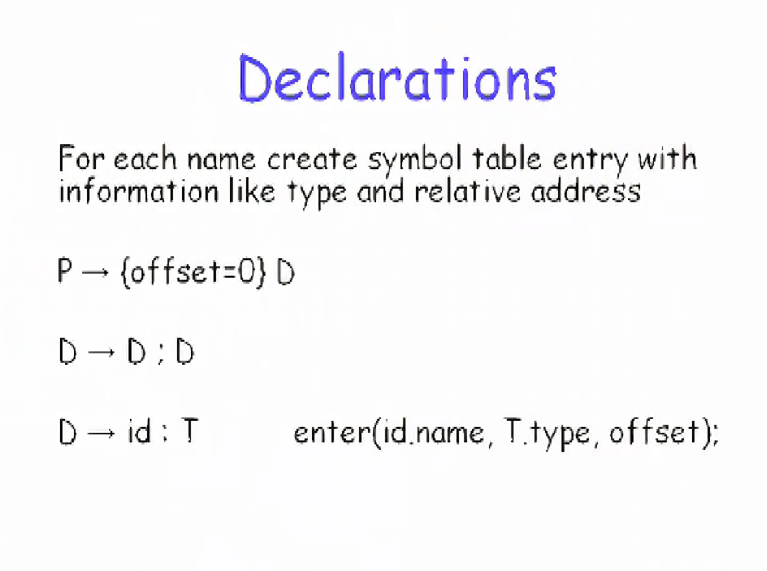
So that is the I will remember and as soon as this function now 4 and +whatever was interior 4+4 yet pushed on the state and offset for I becomes 0 w I am not forcing any information's all this going to be part so now with this one to evaluate the question or this evaluate the clear okay s good so this is kind of program we have know okay so let us see that how do I create the symbol table so this program I want to create a symbol table which of this form .

(Refer Slide Time: 10:38)



So this is my output and what I show the program was input right which is syntax diameter which will make sure that the process the program how come the symbols table so the problem definition here so problem definition is want to know find out how to solve this problem okay the way we solve this is that if I go you do not have respects scopes and I have a level up scope.

(Refer Slide Time: 1:07)



Declarations

For each name create symbol table entry with information like type and relative address

P → {offset=0} D

D → D ; D

D → id : T   enter(id.name, T.type, offset);

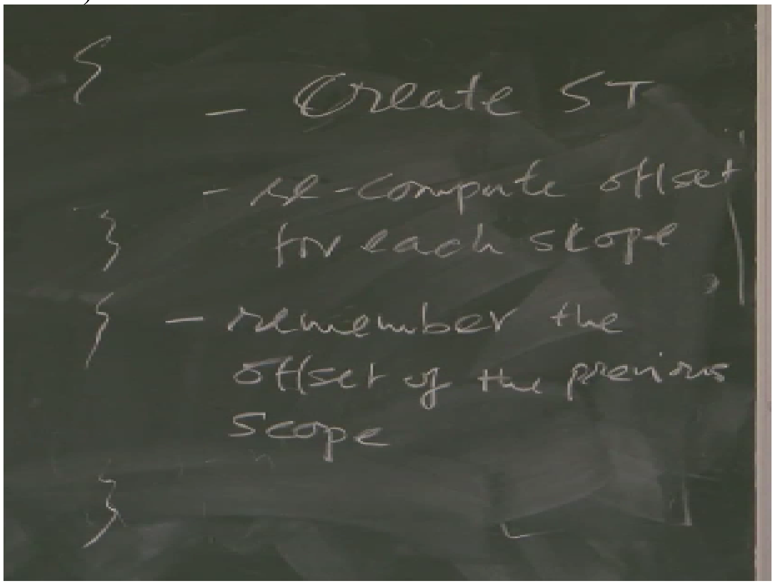
Then we already looked saying that I will start offset of 0 whenever I have declaration I will entry this information symbols table I will identified certain type in offset right and then rest of the variables how I modify this I will do this offset is a sign offset + whatever is a width of a particular time and then so rest of the title declaration I just say what is my time if you recall type information. I already have, when I was doing a time checking I was already use only additional that has come noise the with here.

And some information about the offset this slide has to be recomputed every timer when you entered information symbols there is a only addition thing that will happen rest of the information we had clear okay so what we do now is keep track of local information and what we assume now is that program consists of declarations declaration can be a sequence of declarations it can be declaration of a local variable or it can be declaration of procedure or function.

Okay and what is the declaration of a position function that is an ID for each procedure and function it has its own local declarations and it has set of strings and again I will not state any statement given and it can even worry about I will just be noted for the time but now you can see that I have because in definition that means I can have another procedure inside right and therefore whenever I encounter up to sign a declaration like this I need to create a new symbol table but if I do not get a simple table.

I will keep acting it in the same single table so now a thing you must remember is that I need to create simple tables.

(Refer Slide Time: 13:17)



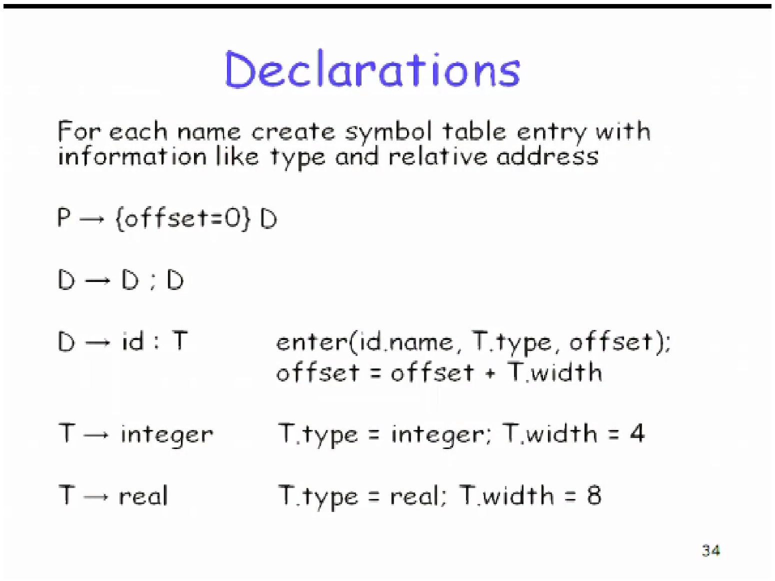
{ – Create ST

– re-compute offset for each scope
}

{ – remember the offset of the previous scope
}

Okay and there is one thing I need to do for other things I need to do for nested scopes I need to recomputed offset for each of the scopes, I also need to remember the scope of the offset of the previous scope okay anything else I need to do come so you also remember we have to say a symbols table and I have this declaration so let us say I have declaration like this.

(Refer Slide Time: 14:24)



Declarations

For each name create symbol table entry with information like type and relative address

P → {offset=0} D

D → D ; D

D → id : T enter(id.name, T.type, offset);
offset = offset + T.width

T → integer T.type = integer; T.width = 4

T → real T.type = real; T.width = 8

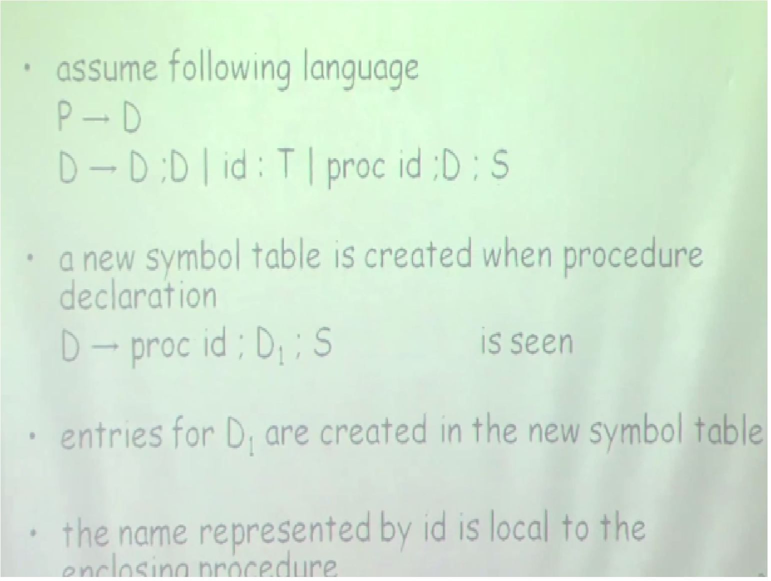
34

It says DIT why we need to say I which symbol table it should be entered right so I need to give the information that entered this particular name of this offset in symbols table how I remember symbol table which so I need to do pointer when I have a symbol table if I have pointer to symbol table most this information this particular symbol table so let me when I say symbol table that means remember the pointer to symbol if I need to remember the pointer from the symbol table I also be remember the pointer to the biggest symbol table because when this finishes for example when this scope finishes and I say that the local various variables of P 1 must be acted in this symbol table but variable L n + 1 must be entered in the symbol table here that means you also need to remember that pointer to previous symbol table.

It will be very simple right now you can immediately see that when I am talking about creating a symbol table having a pointer remembering the old pointer e-computing the offset and remembering the offset okay ,I just had two statics on one I keep the pointer to this very symbol table and on one I keep the offset to the current variable to that particular scope and whenever this symbol table finishes that means this particular scope finishes and I come back I just need to pop this pointer in I need to then be moved offset.

Okay this exist what I am going to do ,So now we will say that when I have this nested scope we will say that entry for deep are not related to the new central table and name is represented ID it is local to the end movie procedures so this ID which is name of this procedure is a local variable to this program but these declarations are inside Id

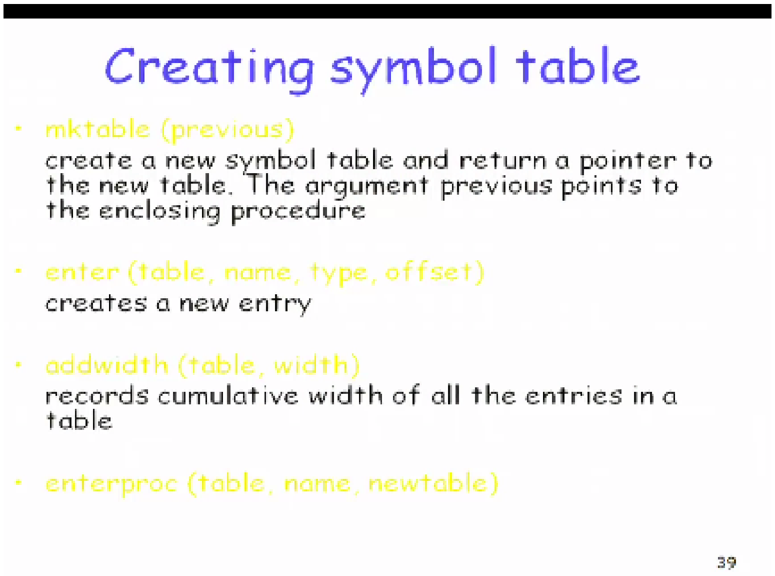
(Refer Slide Time: 16:45)



- assume following language
  $P \rightarrow D$
  $D \rightarrow D ; D \mid id : T \mid proc\ id ; D ; S$
- a new symbol table is created when procedure declaration
  $D \rightarrow proc\ id ; D_1 ; S$ is seen
- entries for D_1 are created in the new symbol table
- the name represented by id is local to the enclosing procedure

So I need to remember the pointer to the symbol table. I need to remember pointer to the symbol table I need to remember the offset in that I only you want to push it in one static with two entries like a static of registers but just information is the same right okay, so let us go through the code and if you still have doubts and we come back okay, so this is the for this is what we want to create him so let us see how do we create symbol table so I have a function which says create a table.

(Refer Slide Time: 17:37)



Creating symbol table

- mktable (previous)
  create a new symbol table and return a pointer to the new table. The argument previous points to the enclosing procedure
- enter (table, name, type, offset)
  creates a new entry
- addwidth (table, width)
  records cumulative width of all the entries in a table
- enterproc (table, name, newtable)

39

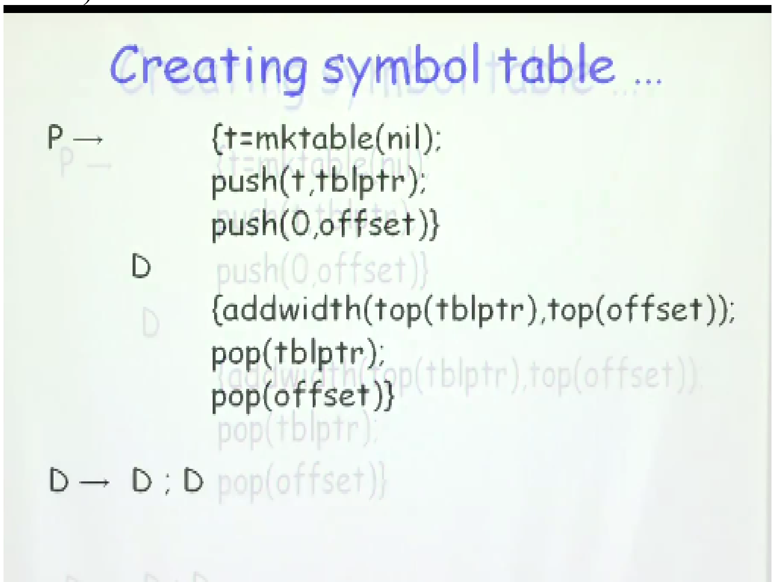
And this also is B pointer to the previous table that means if I am now filled in a table for this it returns just a pointer so create a new simple table intended pointer to the new table and this one I remember and argument previous points to the enclosing to see is so what I am saying is now so when I say that I want to create a simple tables for p1 okay , I will say pass to this function which says create this table pointer to this particular simple table so that when I put this pointer back I remember that one field of this simple table will have to point.

To this particular symbol table okay, so this is what I need to do so now I say that my enter function is going to be I say name, type, offset but I also say that U and D create a new entry and then I also say that at width which says table width so what I am keeping here is so if you recall once feels ahead so here is and the top one was going a pointer to the parent nodes but I left this field undefined and this really is the cumulative a top all the symbols in the scope so foe examples, If some particular scope I have five variables declare then I know that this particular scope.

Okay this information I will also keep ready with so these case cumulative width the all the entries not of the enclosing functions where only the local variable so that information becomes useful when I start designing in my okay , so let us see in presence of these functions or will I do that and one function which says now enter to see your name so this says that within this table enter a procedure within this stable enter name at a pointer to symbol table okay, so when I say that in this function either .

I am going to add a variable or I am going to add a variable of type function and say that this must have a pointer to the new symbol D these are the four functions ID so let us quickly go over the board and then see how it helps me and so when I start now processing my program.

(Refer Slide Time: 20:07)



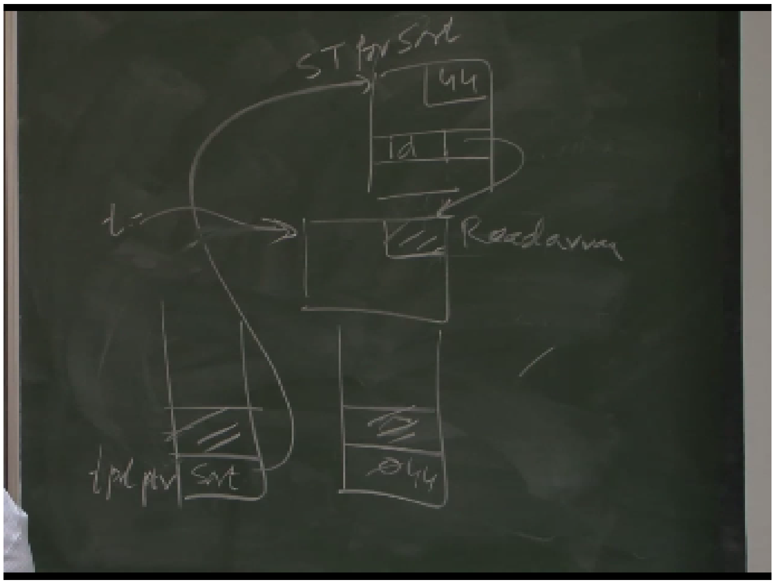
Creating symbol table ...

```
P →        {t=mktable(nil);
            push(t,tblptr);
            push(0,offset)}
      D
            {addwidth(top(tblptr),top(offset));
            pop(tblptr);
            pop(offset)}

D →   D ; D
```

First thing I say is create a symbol table with the point that it has no parent and it is returning a pointer T to me so this is saying that the simple table, I am creating for the current scope does not have a parent node and the pointer to this table is and then I say that I now have a stack in which I remember which is my current symbol table so table points that is which I am going to get this information which says that is the current pointer, so basically what will happen is something like this satisfy now and counter my same program and I first say that the sort is the program at the top level.

(Refer Slide Time: 20:47)

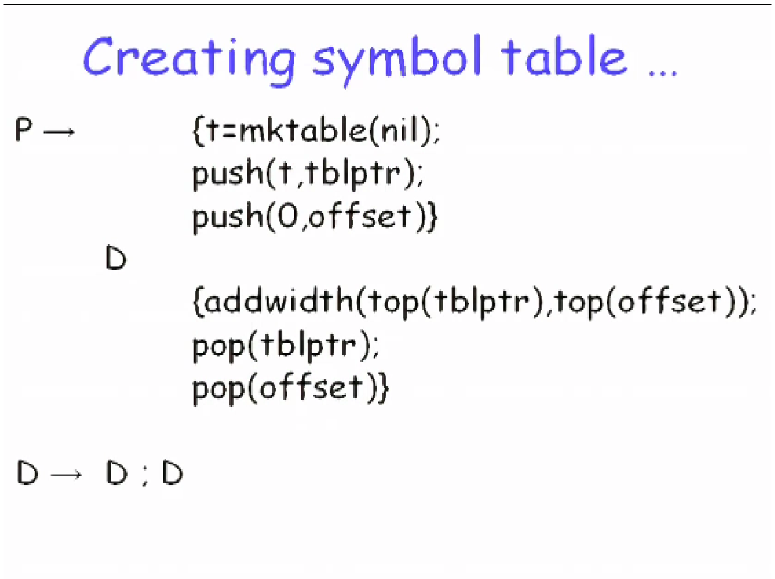


What will happen that I will say create a symbol table for sort so this gives me a pointer and then I have this stack which is table pointer and this has this entry corresponding to source so basically this is nothing but by this function that I created a symbol table and then I push this into the table and then what do I do now I say push zero on the offset so I have another stack in which I keep information about the offset and I push 0 here so earlier object was a global variable not offset is just going on the set and then after this initialization what do I do I have processed all the declarations corresponding to local declarations.

This particular scope and after I finished this what do I say that I am going to add width of all these declarations into the symbol table which is pointed to by the top and whatever is on the top of offset so what will happen is that as I keep ranking variables in this offset you keep changing right and whatever is this final value suppose I had five variables in that so I had available which was added and I had which was four bytes and therefore and that point of time this offset must have been three from 44 and I am going to enter.

This information now 44 so basically they say they add with whatever is being pointed to by this particular entry and this is the value to the enter key that will happen after a half process all the declarations and then I will say now pop this and when I pop this finish to be submitted and I am going to offset all the outer most but what happens to the inner scopes.

(Refer Slide Time: 22:44)



Creating symbol table ...

```
P →         {t=mktable(nil);
            push(t,tblptr);
            push(0,offset)}
      D
            {addwidth(top(tblptr),top(offset));
            pop(tblptr);
            pop(offset)}

D →   D ; D
```

Okay, so far enough scopes are happens is that I have a declaration which is of type will see your end so now suppose within sort program .I have read them right now when I enter the be damaged what will I do when I say pop ID what are the action I need to take I need to create a symbol table so I need to create a symbol table and push the point of here, if so I now see it the symbol table which is corresponding to B that is and this entry okay, and then once I have done this I now say that I am going to push this P on the table and I am going to push zero and offset so I am going to put zero here.

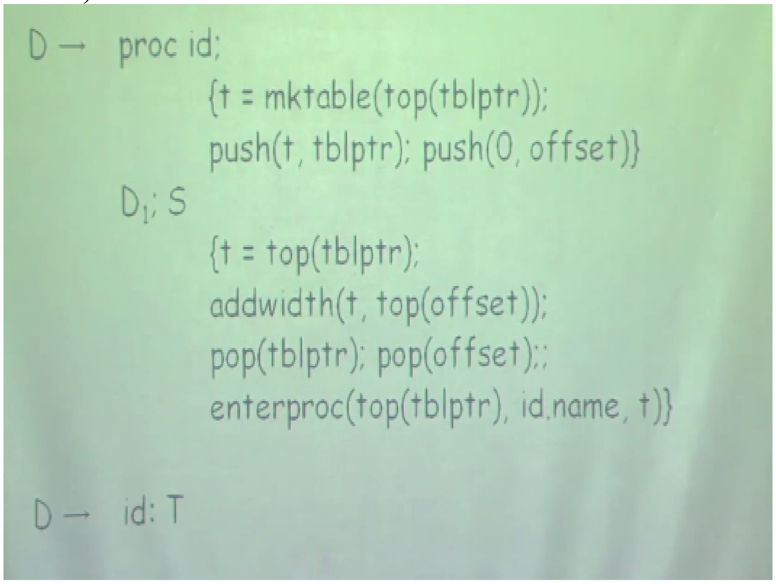
Okay and after that what will happen, I look at the local declarations okay local declarations so we burn out all the local declarations of this procedure but once I have finished processing this what will I say that now P is again this variable copied into variable T so I will again say that T now is a pointer which is pointing to the symbol table and then I say that at which T and whatever is the offset so whatever was the offset of the local variables of this particular scope that we get and enter here right and then I say that now pop both offset and able pointer so that means I am going to form.

This and I am going to pop this that means this particular point that will be offset okay, but I still need to remember pointer to this simple table so what did I do now I need to make an entry in this symbol table where T will be copied right so this is what I do next ,I say that now enter a procedure entry into whatever is pointed to by top of the table pointer now this top of table pointer is pointing to solve so now I say that in this symbol table make an entry which says that whatever is ID name and what ID name here so see your ID right.

So this is name of the procedure and then what is the second point second point says that T so T is now this particular pointer ,so it will start pointing to this right so only additional thing that has happened is that now I can look at not just one scope but I you can look at multiple scopes what multiple scopes by using these new stats so now suppose I have more variables in sort what will happen I have this pointer and I have this offset at the top and I can keep on adding more involved so on.

So if next variable time procedure then I will create a symbol if the next variable is not time procedure here then I enter it here and then keep on changing my makes sense okay, so this is what happens here now what about this when I say that I have now encountered a variable of type B what is the function call I am going to make now.

(Refer Slide Time: 26:05)



```
D →  proc id;
          {t = mktable(top(tblptr));
          push(t, tblptr); push(0, offset)}
     D1; S
          {t = top(tblptr);
          addwidth(t, top(offset));
          pop(tblptr); pop(offset);;
          enterproc(top(tblptr), id.name, t)}

D →  id: T
```

I need to remember that corresponding to this ID this is the type information and where is the offset is in this is back in with symbol table. I make a 19-2 whatever is the top right so then what will happen that now you say enter into the symbol table which is pointed to by the stack this ID name described and whatever is the offset off the stack so instead of low available now I put the same local variable on stack and corresponding to each of the scopes. I have a global variable now this is making sense everything is following me please okay.

So these are the only things and then obviously once I finish this I also need to now change my width offset mean assigned offset plus T bit I will say offset is in the stack therefore top of offset it being assigned topic of sickness p-m desk of the information is going to be yes any questions on this on how now nested symbol tables taking care of both kinds of language where you can have local variables and functions and procedure definitions in inter-merge or you only have local variables forward by procedures both will work in business.

In the second case you find that only keep on hoping everything will you pop any questions from this do not worry okay, any decisions normally your reduction happens in the last statement standardization questions , so after recursion and the control returns to the parent scope you only keep on return you do not right okay, so what I am going to do now is move to stop here today and I quickly returned your width some of the answer scripts and we will also discuss these briefly tell you the scheme and the solutions.

Acknowledgment
Ministry of Human Resources & Development
Prof. Phalguni Gupta
Co-ordinator, NPTEL IIT Kanpur
Satyaki Roy

Co Co-ordinator, NPTEL IIT Kanpur
Camera
Ram Chandra
Dilip Tripathi
Padam Shukla
Manoj Shrivastava
Sanjay Mishtra
Editing
Ashish Singh
Badal Pradhan
Tapobrata Das
Shuubham Rawat
Shikha Gupta
Pradeep Kumar
K.K Mishra
Jai Singh
Sweety Kanaujia
Aradhana Singh
Sweta
Preeti Sachan
Ashutosh Gairola
Dilip Katiyar
Ashutosh Kumar
Light& Sound
Sharwan
Hari Ram
Production Crew
Bhadra Rao
Puneet Kumar Bajpai
Priyanka Singh
Office
Lalty Dutta
Ajay Kanaujia
Shivendra Kumar Tiwari
Saurabh Shukla
Direction
Sanjay Pal
Production Manager
Bharat Lals
an IIT Kanpur Production