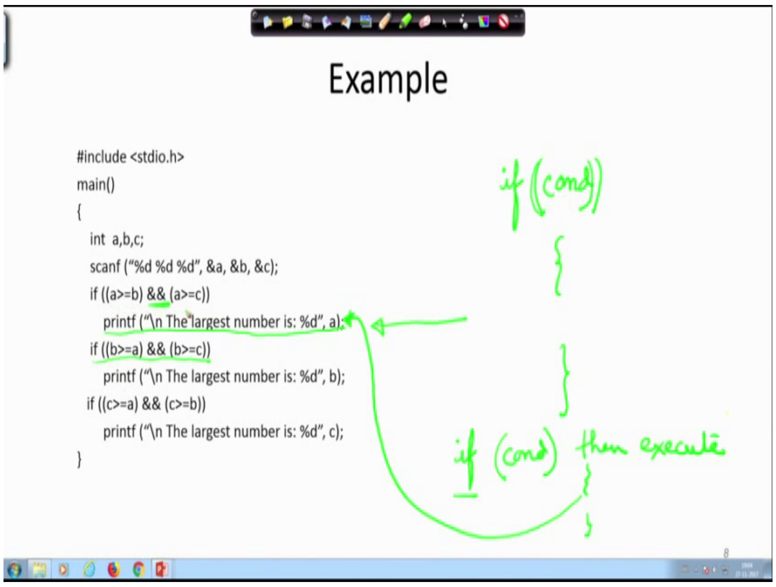
Problem Solving through Programming In C

Prof. Anupam Basu
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 15
Branching : IF – ELSE Statement

We are looking at the IF selection structure where the structure was something like if some condition and then there were some statements, and here is an example of what we are looking at.

(Refer Slide Time: 00:24)



Example

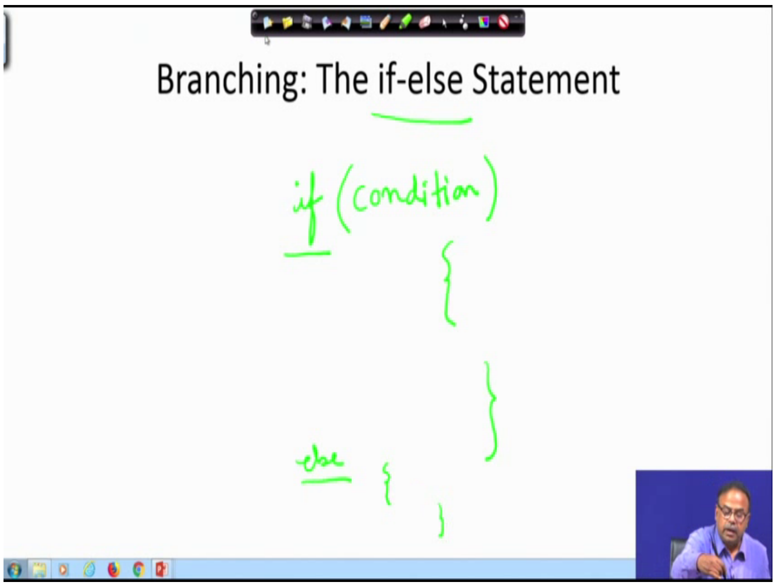
```
#include <stdio.h>
main()
{
   int a,b,c;
   scanf ("%d %d %d", &a, &b, &c);
   if ((a>=b) && (a>=c))
      printf ("\n The largest number is: %d", a);
   if ((b>=a) && (b>=c))
      printf ("\n The largest number is: %d", b);
   if ((c>=a) && (c>=b))
      printf ("\n The largest number is: %d", c);
}
```

So, here you can see that if a is greater than b and also greater than c here is a logical connective. So, then this statement will be executed. So, in English we can just call it like if the condition is true then if the condition is true, then execute a set of statements. So, in this case the set of statements is just only this printf. Otherwise if this condition is true then this statement will be printed. So, this is what we discussed in the last class.

Now, one thing for those who will be writing programs in C you should remember that this condition is always within a parentheses and within that parentheses there can be a composite expression which is here you can see there is a logical expression, here is another logical expression and here is a composite logical expression joining them by AND or it could be by OR etcetera. So, in that way we can form the condition statement and if the condition statement is true then these things will be executed. We will see more examples as we go ahead.

So, let us look at a little more different, a little more extensions of the structure.

(Refer Slide Time: 02:08)

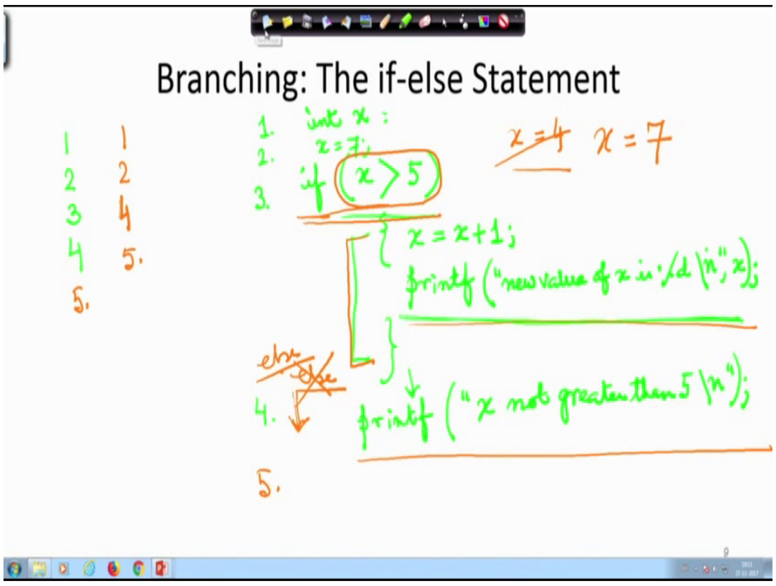


Branching: The if-else Statement

```
if (condition)
        {

        }
else  {
      }
```

Here we are going to see the if-else structure what we actually want to mean is if a condition is true then will be doing these things and if the condition is false then I will do these things. Let me clarify it a little bit more.

(Refer Slide Time: 02:43)



Branching: The if-else Statement

```
1.  int x ;
2.   x = 7;
3.  if (x > 5)
        { x = x+1;
          printf ("new value of x is %d \n", x);
        }
else
~~else~~
4.      printf (" x not greater than 5 \n");

5.
```

~~x = 4~~ x = 7

```
1   1
2   2
3   4
4   5.
5.
```

Suppose I write if x is greater than 5 increment x by 1 and printf “new value of x is percentage d back slash n x” alright. That means, if x is greater than 5 suppose x is an integer I am assuming that x is an integer here. So, somehow somewhere up here I have written int x equal to for int x just int x and later on I have assigned x to be 7 alright.

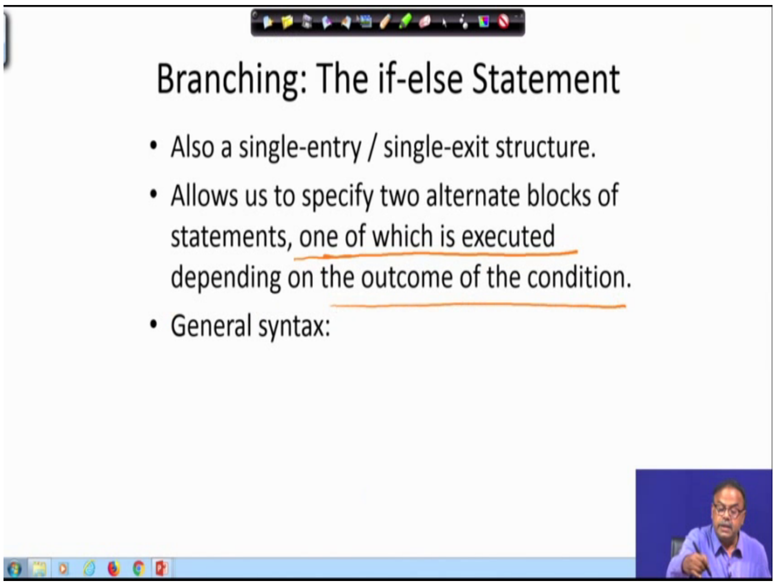
Now, when I encounter this condition this condition always true then I come to this set of statements because this condition is true. So, what will be the value of x now? x was 7 it is greater than 5 therefore, x will be 8 and what will be printed new value of x is 8 that is what will be printed. Now, suppose I write printf x not greater than 5. What I intent to do is if this condition is true then this will be printed otherwise this will be printed that is what my intention is, that is what I want to do. But the way I have written it will actually do something different. You see when the execution will be done you know the execution is normally done in a sequential manner. So, here we will come x is greater than 5, x is 7, so x is greater than 5 this thing will you printed and will come out and come to this next statement. Let me number these statements say this statement was 1, this was 2, this was 3. Now, see I am calling this entire if statement to be a single statement; so 3 and then 4.

So, normally 1 will be executed, then 2 then 3 now 3 will be executed because x is greater than 5. So, this condition is true and then what will be executed 4. So, what will be the output what will be written? Here new value of x is 8 will be printed and here when it 4 is executed again you printed x is not, x not greater than 5, but that was not my intention. So, in order to avoid that I can write a new statement here there was an if, if I write here else this. That means what? If this condition is true then print this otherwise; that means, if this condition is false then do this and then number 5 can be something else and that will be executed. So, what will be the execution? Suppose x is equal to 4, let us try to write down what is execution sequence then 1, then 2.

Now, x is equal to 4 therefore, this condition will be false this condition will be false therefore, this statement 3 will not be executed, this part of 3 will not be executed then we will come to this else statement; that means, if the condition is false then I am coming over here. And so 4 will be executed 4 will be executed, 3 will not be executed completely and then 5 will be executed.

But if I had not put this else then what would have happened in this case if x is 4, in that case this will not be this will not be true only this statement will be printed fine there will not be much problem, but suppose x is equal to 7. And if this else is not there then what will you print it x is 8 x not greater than 5. But if I put the else here if I put the else here I am fairly repeating a little bit more because you lot of students face difficulty in this structure. So, if I put else here in that case this will be x is 7. So, this will be printed and at this else this condition was true. So, this is this part will not be executed, this part will be executed only if the condition was false. So, this contradictory situation would not occur alright.

(Refer Slide Time: 08:43)

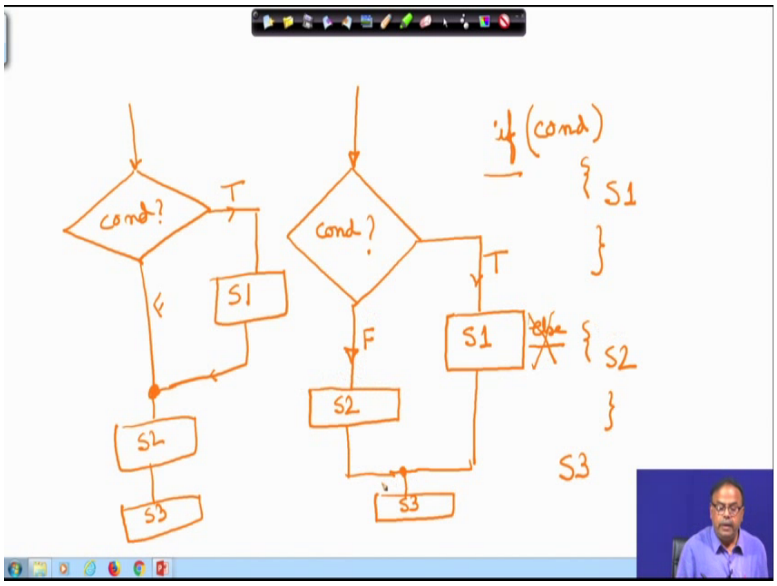


Branching: The if-else Statement

- Also a single-entry / single-exit structure.
- Allows us to specify two alternate blocks of statements, one of which is executed depending on the outcome of the condition.
- General syntax:

So, let us look at some examples of this if statement if-else. It is also a single entry single exit structure it allows us to specify to alternate blocks of statement one of which, one of the which is executed any one of them depending on the outcome of the condition.

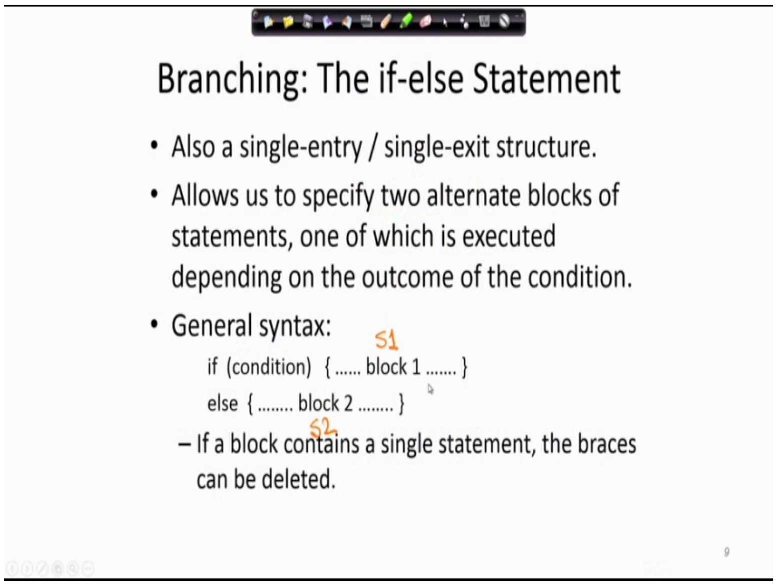
(Refer Slide Time: 09:16)



Essentially what we are trying to say is that I will have a single entry point to a decision box and I will check for some condition I will test that particular condition if that condition is true then true then I will do some state set of statements let us call that S 1, otherwise if it is false then I will carry out some S 2 alright and then I will be meeting may be at some other point which is S 3.

So, this structure I can write as if condition then I do I have name I am not writing the statements I am just writing S 1 else S 2 and then S 3. So, my flow will be either this way and S 3 or this way and S 3. If I had not put this else how would this diagram look like? You have seen this diagram, now if I had not put this else how would the diagram look like. The diagram would look like if here I have got the condition block decision box true then I do S 1 and then I do S 2 and S 3. So, will come at this point if it false then I will also come at this point if it is true then also I will join at this point if this else was not there. But since this else is there then I just separate the out their point of joining alright will see more examples of this.

(Refer Slide Time: 11:53)



Branching: The if-else Statement

- Also a single-entry / single-exit structure.
- Allows us to specify two alternate blocks of statements, one of which is executed depending on the outcome of the condition.
- General syntax:

S1

```
if (condition) { ...... block 1 ....... }
else { ........ block 2 ........ }
```

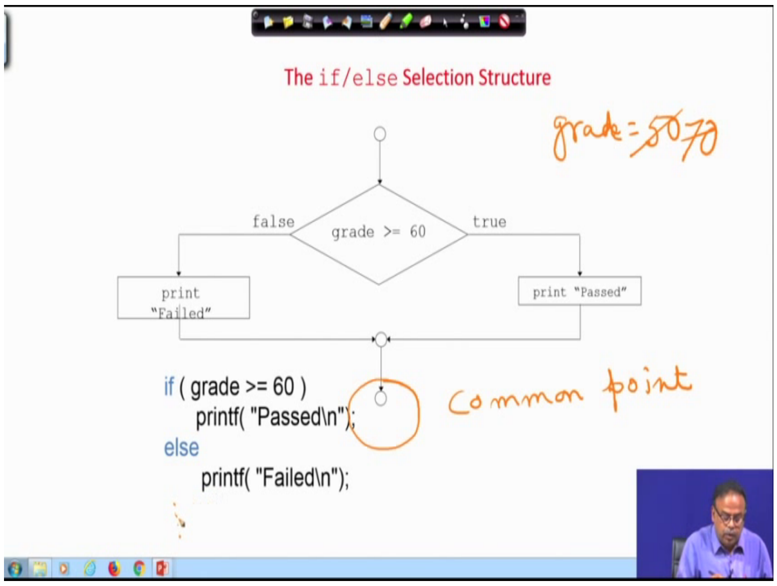
S2

  - If a block contains a single statement, the braces can be deleted.

9

So, the general syntax of this is if condition then block on1,e block 1 is what I was calling S 1 the set of statements else block 2 the other statements alright. If a block contains a single statement then the braces can be deleted I am sorry if the block contains a single statement then this a start essential that I put them, but just for the sake of generality in most of the cases I will have more than one statement therefore, I am putting that in place alright.

(Refer Slide Time: 12:40)



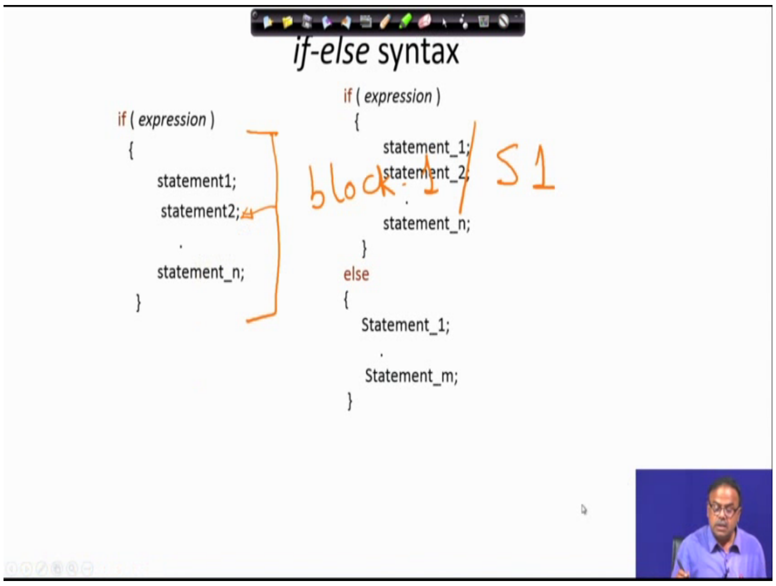
The if/else Selection Structure

```
if ( grade >= 60 )
    printf( "Passed\n");
else
    printf( "Failed\n");
```

So, the diagram that I was looking at I was explaining that sort of diagram. Suppose the grade is 60, if the grade is greater than equal to 60 suppose somebody's grade is or marks is 50 then at this point the condition is false this part will be executed, if the grade is 70 that is this condition is true this part will be executed. Then I will come to this common point, this is a common point which will come.

So, this I can write as if grade is 60 see here there is only one statement. So, I did not need not put the I have not put the braces here print sorry printf passed else printf failed and then whatever I had to do I will do it here and then whatever was following that will follow here. So, let us come to. So, I hope this is clear.

(Refer Slide Time: 14:03)



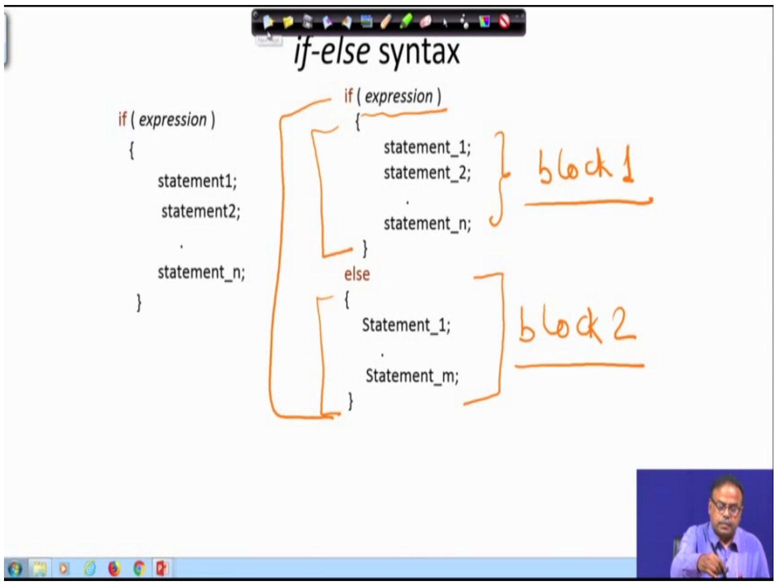
if-else syntax

```
if ( expression )
{
        statement1;
        statement2;
        .
        statement_n;
}
```

```
if ( expression )
{
        statement_1;
        statement_2;
        .
        statement_n;
}
else
{
    Statement_1;
    .
    Statement_m;
}
```

So, the syntax of if-else let us look at little bit more here I have said block 1, block 1 means or S 1 whatever you name it. This is block 1 or I had name it as S 1 the same set of statements where their number of statements each separated by a semicolon. Please note suppose there n statements here separated by a semicolon.

(Refer Slide Time: 14:50)



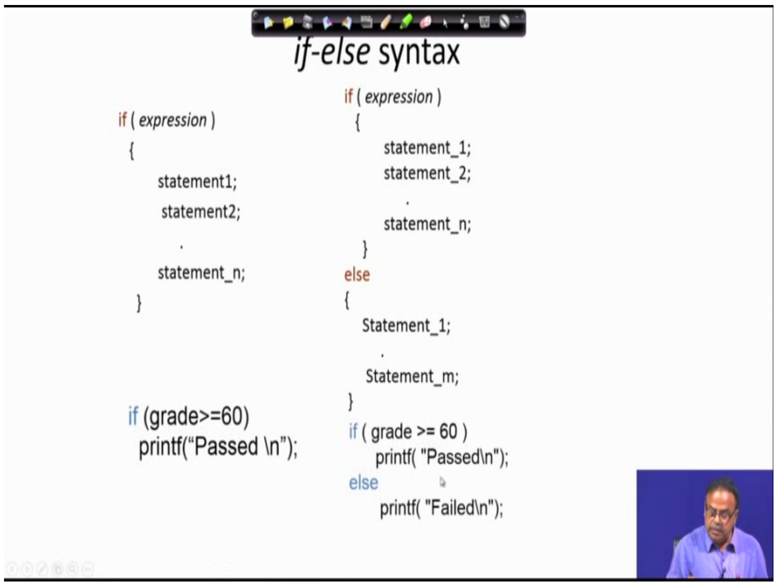
if-else syntax

```
if ( expression )
  {
      statement1;
       statement2;
       .
      statement_n;
  }
```

```
if ( expression )
 {
     statement_1;
     statement_2;
       .
     statement_n;
 }
else
{
   Statement_1;
     .
   Statement_m;
}
```

Now, next I come to this if I want to do else also if expression statements 1 to n, else state these statements 1 to m. So, this is my block 2 and this is block 1. So, here we can see that based on this expression which is a condition evaluation if this condition is true I will carry out these statements else I carry out these statements. Note that this individual statements have got semicolon and this is a whole body of the else statement this is the whole body of the if statement. So, composite if this entire thing we call an if-else structure.

(Refer Slide Time: 15:39)



if-else syntax

```
if ( expression )
  {
      statement1;
       statement2;
       .
      statement_n;
  }
```

```
if ( expression )
 {
     statement_1;
     statement_2;
       .
     statement_n;
 }
else
{
   Statement_1;
     .
   Statement_m;
}
```

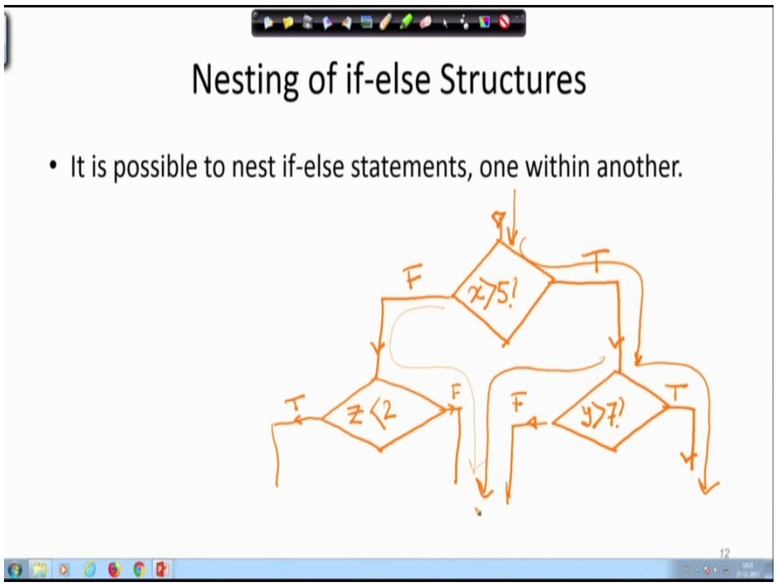
```
if (grade>=60)
  printf("Passed \n");
```

```
if ( grade >= 60 )
    printf( "Passed\n");
else
    printf( "Failed\n");
```

So, if grade is 60 printf passed else print failed.

Now, this next we are coming to a little more complication of the same thing it is the next thing of if-else structure.

(Refer Slide Time: 15:44)



Nesting of if-else Structures

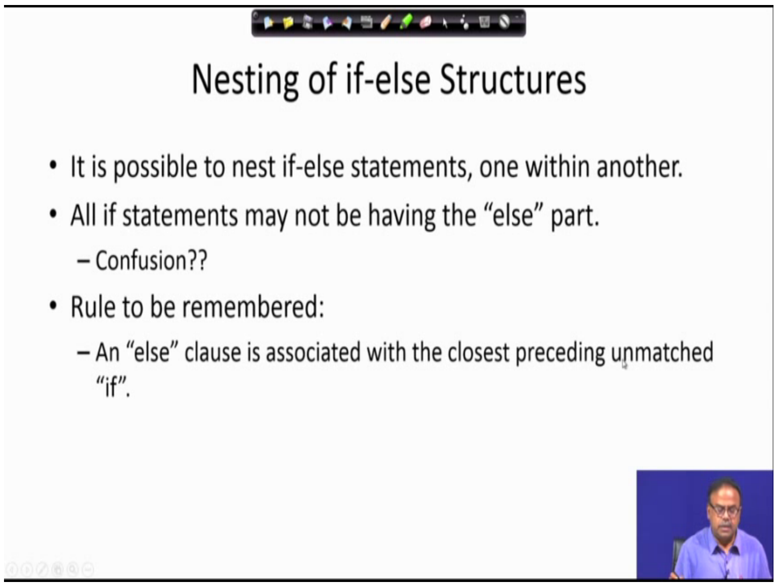
- It is possible to nest if-else statements, one within another.

Now, why this if-else what are we trying to achieve we are achieving branching. That means, we are carrying out a sequential flow we are carrying out a sequential flow and from there we make a decision box and then we branched either in this direction or in this direction.

Now, suppose here in this direction it is true and this is false again I come and make a decision. It can be, if x is greater than 5 then I come here and y is greater than 7 then I do something here it can be x is greater than 5 I follow this path, but y is not greater than 7. So, I follow this is true this is false I follow this false path. If x is not greater than 5 I come here and suppose I check z less than 2 true or false there can be 2 outcomes. So, suppose x is 4, not greater than 5 I will come through this path, now z is 3, z is not greater than 2, less than 2 therefore, I will follow this path. So, you see the path that we follow that is a branching that we do can have the path that we follow can have more than one decision box say x is greater than x is 7. So, here I was coming in this path then I follow this path because x is 7 this condition is true and I find that y is 8 so I follow this path alright, if y was 6 then I would have followed this path.

So, in the path that the program follows there can be more than 1 decision box. So, that is what we mean by nesting of if-else structures.

(Refer Slide Time: 18:05)



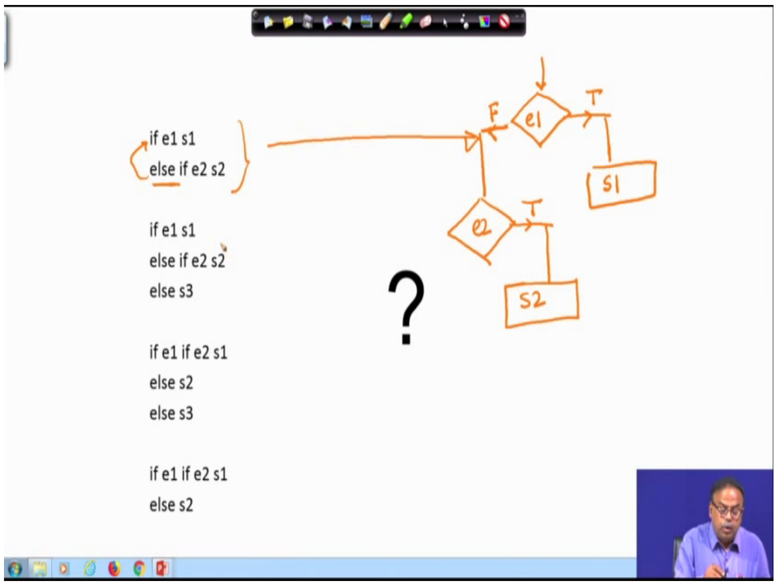
Nesting of if-else Structures

- It is possible to nest if-else statements, one within another.
- All if statements may not be having the "else" part.
  - Confusion??
- Rule to be remembered:
  - An "else" clause is associated with the closest preceding unmatched "if".

That means nest one if-else statement within another. Now, all statements may not have the else part. Now, rule to be remembered is an else clause is associated with the closest preceding unmatched if really confusing what do I mean by this.

Let us look at this how will this be executed.

(Refer Slide Time: 18:30)

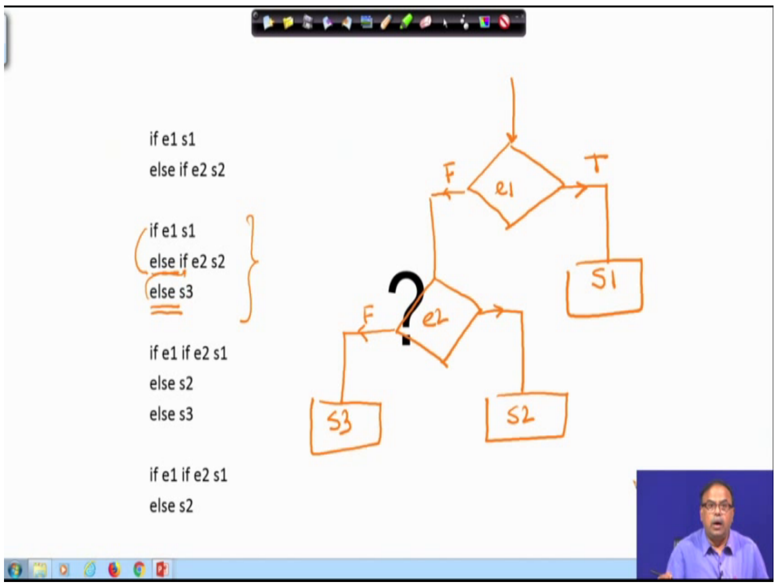


```
if e1 s1
else if e2 s2

if e1 s1
else if e2 s2
else s3

if e1 if e2 s1
else s2
else s3

if e1 if e2 s1
else s2
```

Here e1 means some expression, if e1 is true then s1 else if e2 then s2. Now, with whom does this else go? This else certainly goes with this preceding if. If I just quickly draw this that flowchart corresponding to this how does it look like? I come here and x evaluation, evaluate the expression e1 on true I do s1, on true I do s1, on false here is false alright I come here and there is another if here. So, I again check e2, if e2 is true then I do s2. This statement actually means this graph. If e1 is true then I will carry out s1 otherwise else is there. So, this else is linked to this if then I do this one.

What about this? Let us erase this for a while and how would we interpret this one. Here you can see the two elses. With whom is this else paired? This else is paired with I have written it in a bad way, this else is paired with this if.

(Refer Slide Time: 20:31)

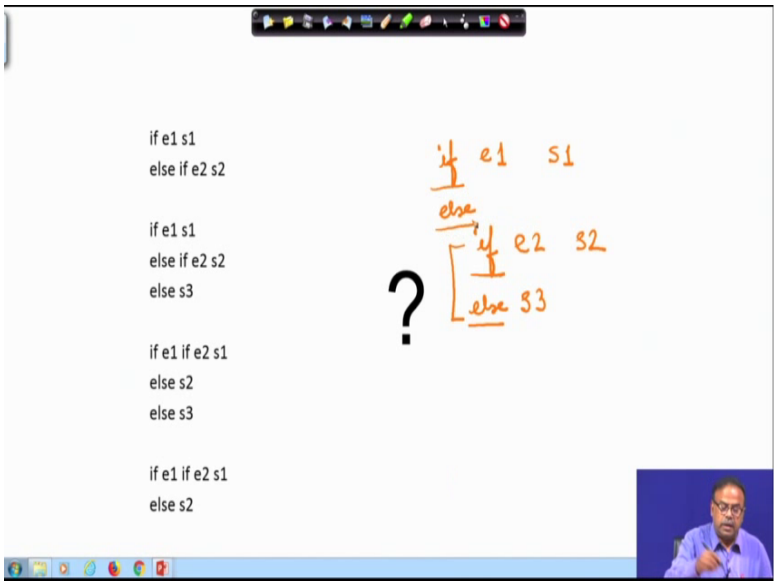


```
if e1 s1
else if e2 s2

if e1 s1
else if e2 s2
else s3

if e1 if e2 s1
else s2
else s3

if e1 if e2 s1
else s2
```

So, this means again I check e1, e1 is true so I carry out s1 else false; that means, this else is for this decision box I check here e2, e2 is true so I carry out s2 and else I carry out s3 this is what I am trying to do.

Now, here I would have preferred to write it in a better way. That is why if you recall in a earlier lecture we had talked about a good program writing practice that is indentation.

(Refer Slide Time: 21:35)

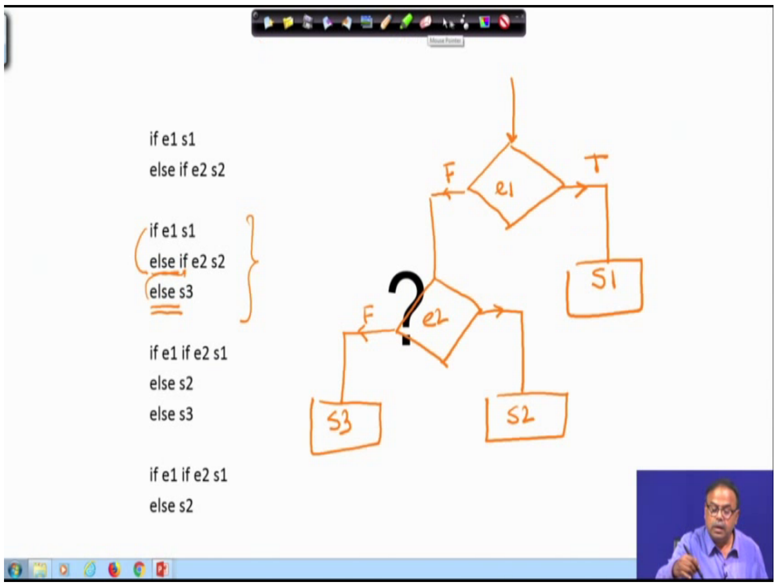


```
if e1 s1
else if e2 s2

if e1 s1
else if e2 s2
else s3

if e1 if e2 s1
else s2
else s3

if e1 if e2 s1
else s2
```

I would have like to write it in this way if e1 then s1 else if e2 then s2else s3 which makes a very clear that this else is corresponding to this if and this if this else is corresponding to this if right.

Let us look at the third scenario what does it mean. If e1 if e2 s1 how would I draw that?

(Refer Slide Time: 22:13)

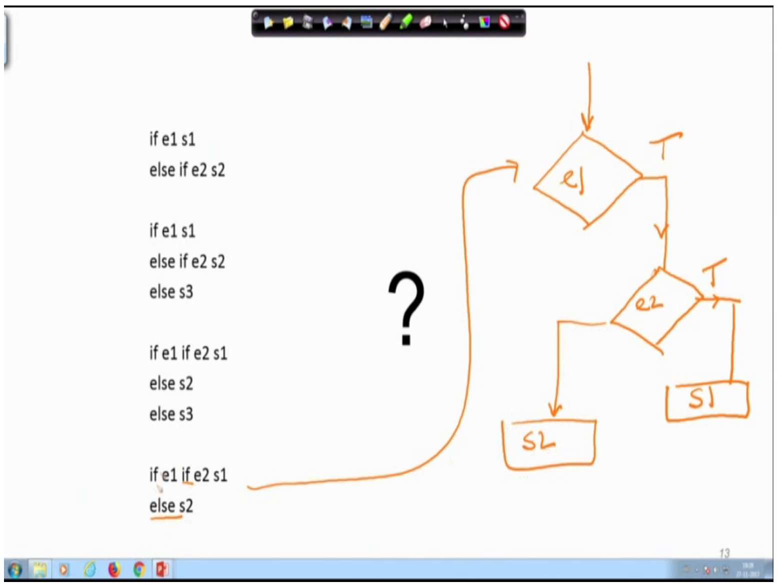


```
if e1 s1
else if e2 s2

if e1 s1
else if e2 s2
else s3

if e1 if e2 s1
else s2
else s3

if e1 if e2 s1
else s2
```

I draw that like that I take e1 if e1; that means, if e1 is true I immediately come here and the statement starts with another if. A nested if, this is known as the nested if one if within another there is one if statement there is another if statement within this, this is knows as nesting alright.

So, here sorry over here if e2 s1. So, immediately I am there is no else here by the way. So, if e1 then I come here then if statement is again there therefore, there is another decision box where I am checking the condition e2 and if e2 is true then I will follow then I will carry out s1 else s2, which else should it be here or there. Now, this is something this else is pairing with a preceding if, so it should be false will be s2. Now this else is therefore, covered this path is covered. So, this else can only mean this one and s3 will be because this is already covered right. So, this is the nearest one to this clear. So, if we go to the earlier slide. Let me go back to the presentation.

(Refer Slide Time: 24:17)



if e1 s1
else if e2 s2

if e1 s1
else if e2 s2
else s3

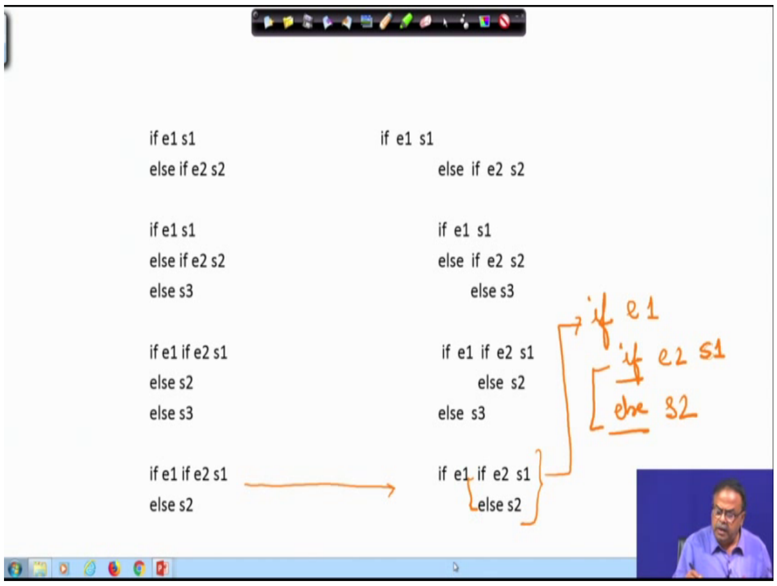
if e1 if e2 s1
else s2
else s3

if e1 if e2 s1
else s2

So, here this one how would this one actually look like? How will this one look like? There is something still there. So, how will this one look like; coming straight here if e1 if the condition is met then I again test here if e2 this is true this is true then I come and do it here s1 and else this else is again with this if. So, it will be here s2 and I have not specified this one here alright.

So, if you put a little bit of mind here. So, you can see many different other scenarios that can come and we can proceed accordingly.

(Refer Slide Time: 25:16)

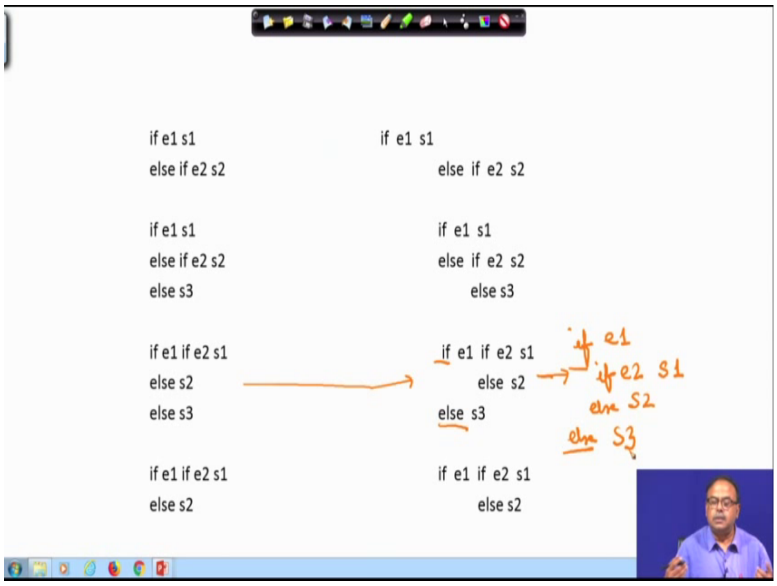


```
if e1 s1                    if e1 s1
else if e2 s2                   else if e2 s2

if e1 s1                        if e1 s1
else if e2 s2                   else if e2 s2
else s3                             else s3

if e1 if e2 s1                   if e1 if e2 s1
else s2                              else s2
else s3                         else s3

if e1 if e2 s1                  if e1 if e2 s1
else s2                             else s2
```

Let us see here. This one as I was saying it will be nicer to write it in this way because here you may feel there two elses two if who goes with whom. Now, here if I write it in this way if e1 s1 else; that means, this if-else else if e2 s2. Here if e1 s1 else if e2 s2 else s3. So, this else is therefore, with I am sorry I think I will have to go back to a couple of slides you see and else clause I would like to be corrected a little bit and else clause is associated with the closest preceding unmatched if, any if that is not there that should be it should match with that. So, that is what I explained in these examples.

So, it is a much better for example, just look at this how would I like to like write this. If e1 I would not prefer this also, I would prefer that I write it in this way. If e1 then if e2 s1 else s2 that becomes very clear this if-else and else are very clear. Say for example, here also this is better, but still I would prefer to write this one as this one, this is one way I can write if e1 if then I mean this is the part of if, if e2 s1 else s2 and under this if that is what is been done here else s3.

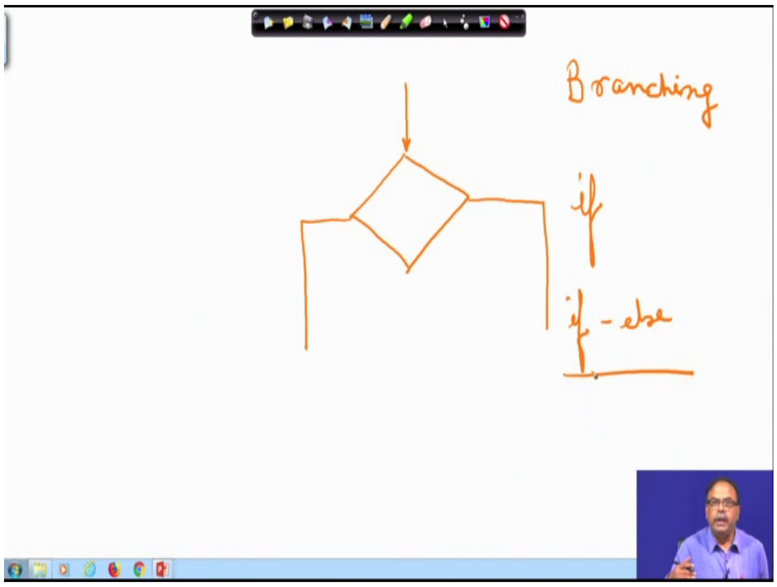
(Refer Slide Time: 27:17)



```
if e1 s1                    if e1 s1
else if e2 s2                      else if e2 s2

if e1 s1                           if e1 s1
else if e2 s2                      else if e2 s2
else s3                                 else s3

if e1 if e2 s1                     if e1 if e2 s1      if e1
else s2                                 else s2    →     if e2 s1
else s3                            else s3                else S2
                                                        else S3

if e1 if e2 s1                     if e1 if e2 s1
else s2                                 else s2
```

So, what we have learnt today is a nesting of another structure. What we are discussing now is how we can have a better or more versatile flow.

(Refer Slide Time: 27:55)



Normally it is the sequential flow, but based on the decision points I may have to take different paths which is known as branching right and for branching or selecting the proper path the structure is if. Along with that we have learnt today if-else structure and we have seen how if-else structure can be nested to give rise to some better more versatile scenarios. We will continue with this in the next lecture.

Thank you.