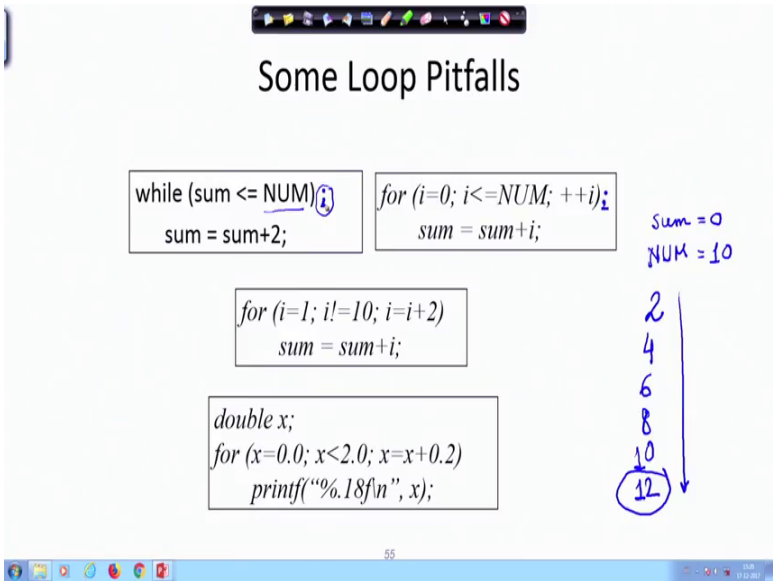
Problem Solving through Programming In C
Prof. Anupam Basu
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 25
Example of Loops (Contd.), Example of For Loops

So, we have seen the application of the constructs of if then and if else as well as while, do while and for constructs. Till now you have seen examples in of while and do while, will see more examples for loops for example.

(Refer Slide Time: 00:47)



Let us also see another application for the application of for loop. Say for example, I want to print I want to add 20 numbers I think we have seen such examples will see more interesting examples a little later. But before that let us look at some common errors that take place often unintentionally in writing loops and that gives rise to a number of logical problems in a program.

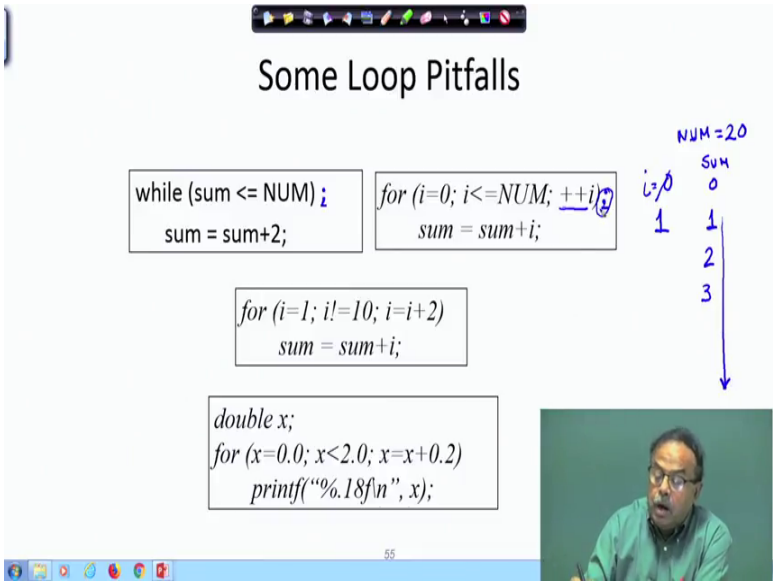
For example, let us look at this. Here you can see the first line what will happen, the intention was that while sum is less than equal to NUM; that means, it is less than equal to sum a particular value may be 20 we are adding sum plus 2. What does this mean? Let us try to understand this example forget about this, forget about this part. What would what is the intention of doing this? That means, suppose NUM is 10 and sum is 0. So, while sum is less than NUM I will be adding sum and 2. So, sum will be, sum will be 2

and then sum is still less than num so again it will be 4, sum is still less than NUM it will be 6, still less than NUM it will be 8, still less than equal to NUM it will be 10, then still less than equal to NUM it will be 12 and then when it goes there it will stop. So, what will be the sum? Sum will be 12. But that was my intention of the program.

But unfortunately I have put a semicolon here. What does that imply? That implies that the entire while statement ends here; that means, while sum equal to NUM do nothing all right. So, that is the end of the statement. So, nothing has been specified there and whatever some was there suppose sum was 0 sum will perpetually remain less than 10 or NUM and will go on forever. So, this semicolon should not to be given because the while statement is actually extending up to this point up to this point that is the whole statement.

So, next example we take is this one, for i assign 0, i less than equal to NUM plus plus i, forget about, again about this.

(Refer Slide Time: 04:08)

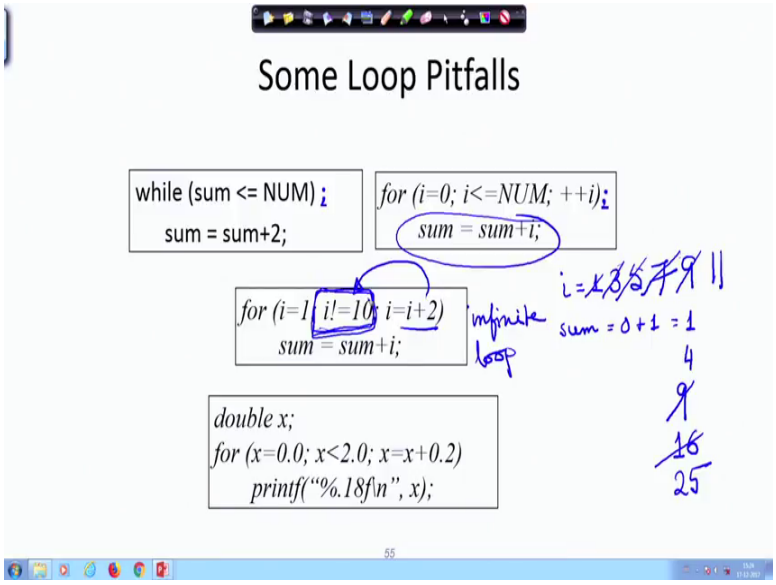


What is the intention of this program? What will it do? i is 0 all right and NUM was say something like 20 then sum will be added to i. So, sum will be 0, then i will be incremented, so i will become 1 less than NUM it will be added. So, sum will now be, sum was 0 so sum will now be 1, here was sum, sum was 0, sum becomes 1, then sum becomes 2, sum becomes 3 like that it will go on all right.

However, since I have put a semicolon here this part is not a part of this for statement. Consequently this loop is a non loop nothing is being done here and for i equals 0, iless than equal to NUM i plus plus do nothing all right. So, nothing will happen here this statement will not be executed.

Here is another type of pitfall where let us study this for i equal to 1, i not equal to 10, i assigned i plus 2. What will happen to this loop? Can anyone guess?

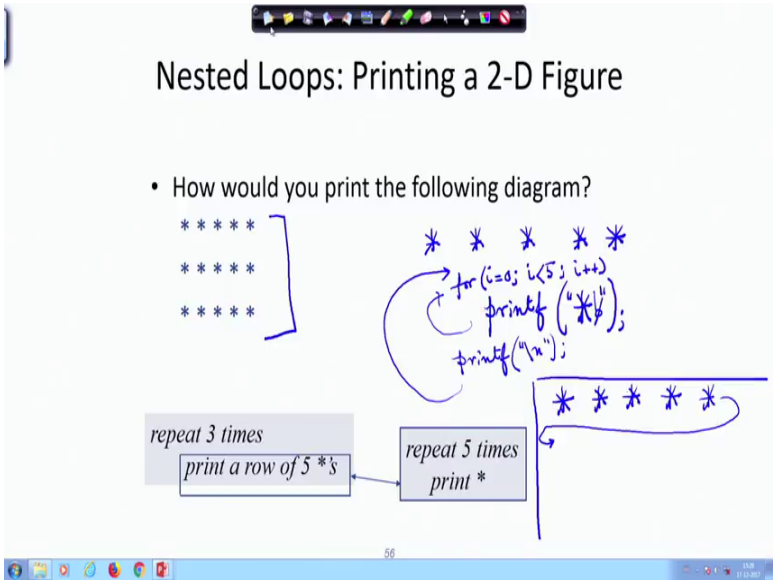
(Refer Slide Time: 05:42)



i is 1, so sum has been computed sum is sum plus i sum was 0. So, 0 plus 1 sum is 1, then i is incremented to 3, then sum equals sum plus i. So, sum becomes 4 then this is incremented to 5. After each incrementation I am checking with this condition, so i is 5 not equal to 10 fine. So, I will add 5 with this, it will become 9, then becomes 7 i is change to 7, I check with this still not equal to 10. So, 7 is added to this 16, this becomes 9 still not equal to 10. So, then 9 plus; 16 plus 9 maybe 25 and then this is incremented to 11 because i plus 2, still it is not equal to 10. So, what will happen? It will go on it will never be equal to 10. This means as long as i is not equal to 10 you will go on. So, this will be a case of any another infinite loop all right.

Just as in this case this is a null statement this will be done and then this statement will be done only once. Here it will be an infinite loop because this condition will never be made. So, these are some points where we should be very careful about.

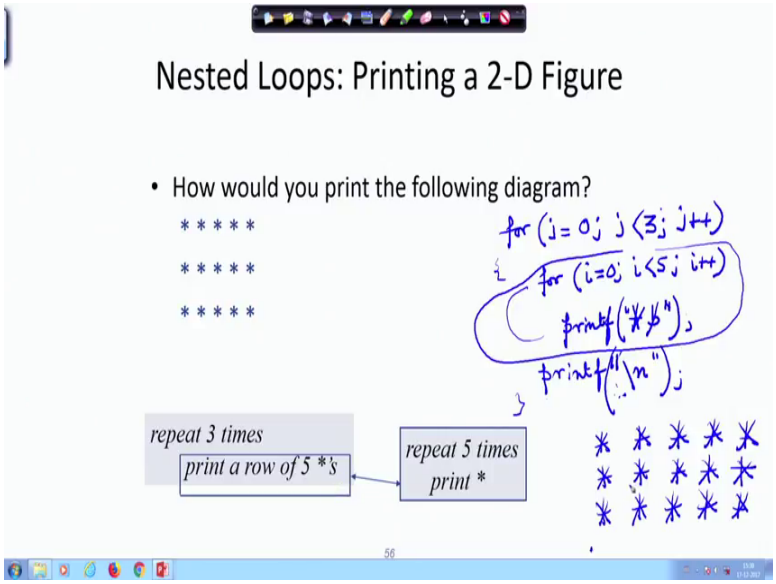
(Refer Slide Time: 07:40)



Now, let us have a very interesting program. We want to print we want to print, we want to print a pattern like this. We want to print the pattern like this say 5 stars in a row and 3 such rows, this sort of pattern. How can I do that? My algorithm will be, I want to print, I want to print a row of stars. How many? 5 stars. So, how can I print 5 stars in a row? If I just write one statement printf, star I do not give a backslash n; then what will be done? A star will be suppose this is my screen all right a star will be printed and if I put it in a loop say for i assign 0, i 5 I want to do, less than equal to 5 i plus plus and I do this then what will happen, i 0, so once it is printed then i 1 once it is printed, again in the same line then again another one.

So, since I am giving a gap what I can do here I can keep a space here star and then a space I show space by blank. So, exactly a star in the blank will be printed. So, i 0, i is 1, i is 2, i is 3, i is 4 and then it is incremented and checked is 5, so less than 5 it will not happen then. So, this will be a loop after doing that. So, in a loop I will be printing one row then I will give printf I have to come to the next line. So, I will simply give a backslash n; that means, I will come to the next line. And this again loop I will carry out how many times? 3 times because I need 3 rows. So, what should I do? I should do this again loop this 3 times so how should I write it, how would that look like now.

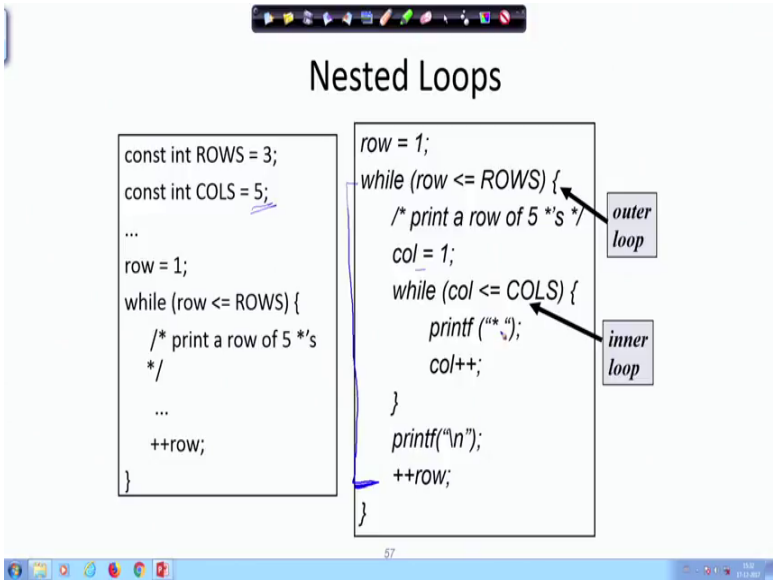
(Refer Slide Time: 11:11)



It should be something like this for j, j is another variable assign 0, j less than 3, j plus plus for i assign 0, i less than 5, i plus plus and then here printf star followed by a blank and then the quote no backslash at the end of this. So, after this loop is done then I will do printf backslash n backslash n. So, this part will be repeated 3 times and in this part it will be this one will be done 5 times. So, star star star star star will be printed. Then we will come to the next line and here this is my next for loop. So, I come to printf and again do the same thing this part start start start start star 5 times, this printf by this loop. This is printed row, printing a row by 5 times. And then again I will come and do a backslash n, I come here and do the same thing 3 times and then come here backslash n and then stop.

So, print a row of 5 stars and repeat therefore, I am repeating this print star 5 times in a loop. So, that is a very nice interesting application of for loop. I hope you have understood this.

(Refer Slide Time: 13:38)

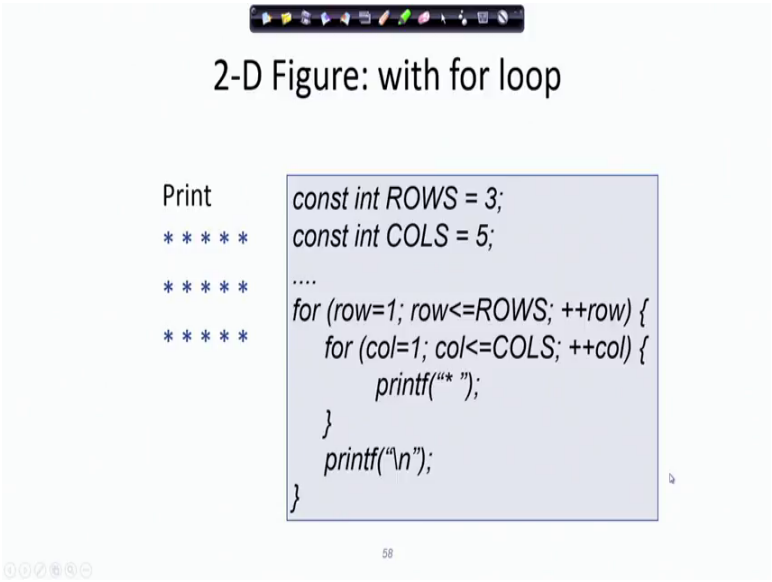


So, here that is exactly what I have I was showing you. Look at this, here the number of rows and columns have been made little flexible rows 3 columns 5. Now, row equal to 1, while row is less than rows print a row or 5 stars. This I have done it is the for, here it is shown using a while.

So, let us see while whether you can understand this also with the while. Row is 1, now row is less than rows; how many rows we will do, row is less than rows that means, as long as it is sorry as long as it is 3 rows will print a row of 5. And how do I do a row of? I will that we have already shown that how we do it and then we increment the row all right. So, here while row is less than rows print a row of 5, printing a row of 5 is done through in this manner. So, this is the outer loop, this is outer loop. Column is 1, while column is less than columns, so 5 columns 1 2 3 4 5 while column is less than column printf star and blank and then column is incremented.

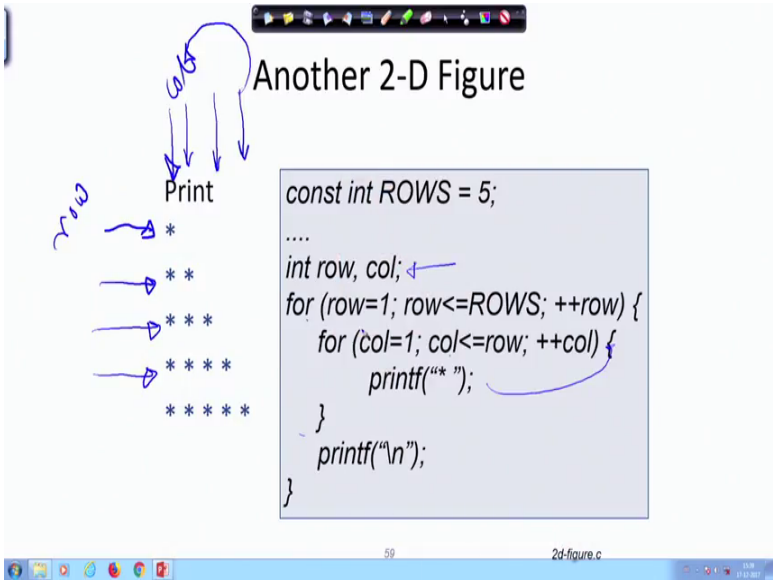
Now, since I am doing it in a while it is done in this way. I have already shown you in the earlier this thing how we can do it with for. I can do the same thing with for right, the same thing with for and here it is being shown how it can be done with a while. And then I print of n and do this. You can try to understand this again yourself.

(Refer Slide Time: 15:54)



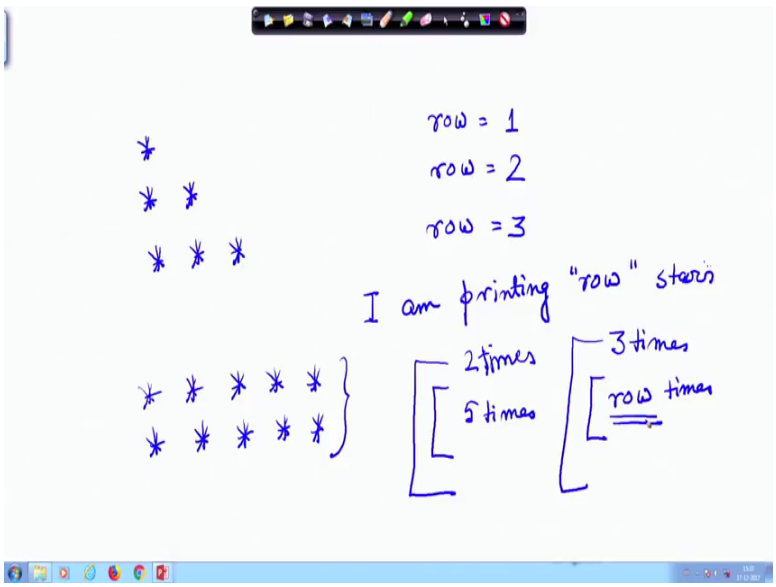
Next, say here this being done again in the way that I had written using for. Here only 3 and 5 these things are variable for row equals 1 to row less than equal to 3, plus plus row. Here it is plus plus row; that means, first is incremented. Then column is less than equal to columns. Why it is less than equal to, while if you remember when I was doing it here when I was doing it I had less than 3 less than 5, less than 3 less than 5, but here it is being less than equal to 3 less than equal to 5 why because I started my index with 0 and here I am starting my index with 1, all right. So, this you should be very careful and you should always hand trace your program and see whether you have done it correctly if there is a little bit of confusion because this is very, this very important and you should be very careful about it.

(Refer Slide Time: 17:19)



So, same thing that I have shown is written again here. Another 2D figure this a little more interesting. First row we print 1 star, second row we print 2 stars, third row we print 3 stars, 4th row we print 4 stars and then 5 stars. So, how many stars I will print that is also variable. So, if we think about that how many times I will print in a row that is also a variable. How many times I am doing this?

(Refer Slide Time: 17:59)



So, for the first when a row is equal to 1 then I am printing 1 star, when row is equal to 2 I am printing 2 stars, when row is equal to 3 I am printing 3 stars. So, every time I can

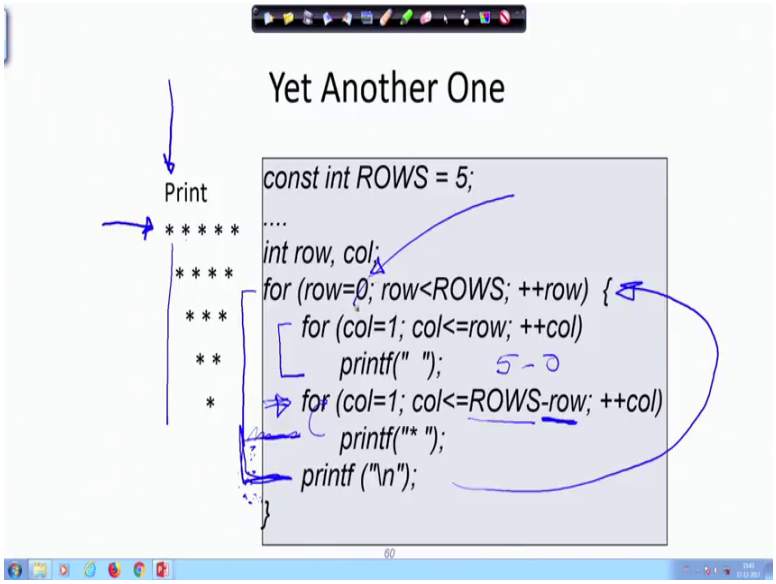
also say that I am printing row stars, I am printing row stars, row number of stars therefore, how many times I will do in a loop in the inner loop. You could see that when I we had drawn this thrice or twice whatever there was here, there is an inner loop 5 times and an outer loop that was doing 2 times right. Now, here what will change it to is inner loop row times outer loop maybe 3 times. So, this is variable.

Now, let us see how we can program it. Constant integer rows is 5 that I have not made variable there are 2 integers row and column. For row equal to 1, I start with rows row as an index and row less than rows less than 5 plus plus row. What do I do? Column equals one I will do up 2 column less than equal to row. So, first row column 1, column is less than equal to the only ones I will print then plus plus column, so column becomes 2, but what is my row? Row is 1 still I am pointing at this row, row is 1. So, I will do printf. Now, the value of row becomes 2 as I increment, but then the column this is a column this is row the column will be less than row because row has become 2. Now, it is pointing to this row, this row. So, column being less than that I will come out of the first row.

Second row, what will happen? Column will start from one and row is 2. So, column less than row I will print once, I will go back here column is becoming 2 and column is still equal less equal to row all right not less than, but equal to row therefore, I will again print here then I will come back here and now column is 3 column is 3, but row is 2 therefore, I will not print any longer will come out of this loop and go here. Then low will be incremented here all right, row has been incremented here. And, now again column is initialized to 1 for the third row first it is printed incremented column comes here all right then columns column comes here, so column is 3 still less than equal to row. So, I print the third one and then it is incremented, so the row becomes 4, but my column is sorry the column becomes 4, but row is 3 therefore, the column is I go out of the loop and again column is initialized to 1 and my row is incremented to this.

So, that is how it is done. You please look at it more carefully and you will have to understand it and this will give you a very clear idea how a nested loop is working. So, this is one example.

(Refer Slide Time: 22:52)



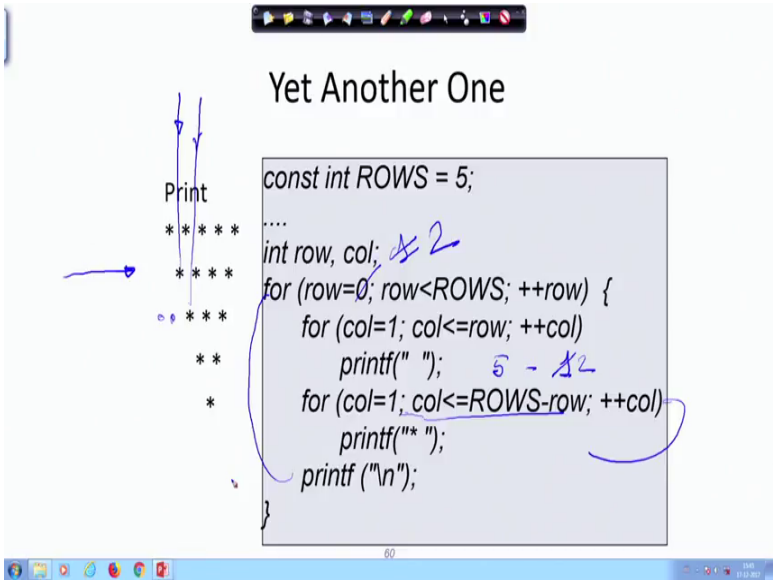
Here, what we are trying to do just think of this figure. First row is 5. So, row is this, column is here how long shall I print in the columns keeping the row fixed. What will be my logic? The logic will be well here I started row with 0 and less number of, less than number of rows, rows is 5 that is not that important. Here let us look at this, first I will do 5, then I will do 4, but there is another one. For the second row I am shifting one space I am shifting this should have been aligned and then I am shifting and giving a space and then doing it. So, gradually it is being shifted.

So, let us see what is being done. Let us look at the first for loop here this is the outer for loop, let us outer for loop is up to this, outer for loop is up to this and let us see what is happening. Row is 0 to number of rows less than. So, I have started with 0. So, I did not make it less than equal to it is less than less than 5 I will do this number of times. Internally what am I doing this doing here column is 1 and column is less than row because less than equal to row because row is 0, row is 0 and column is 1. Look at the trick here, the trick that has been applied is column has been in is starting with a value 1 and as long as column is less than row I am printing blank.

So, how many blanks should I print for the first column? 1 blank then column less than. Now, here from the first column look at this point. Column equals 1, 1 column less than equal to rows, rows is 5 as long as column is less than equal to rows minus row. What is my row? Initially my row is 0, so rows minus row is 5, it is 5. So, column is one column

is less than equal to 5 plus plus, so I do printf, I do a printf and I go on doing this as long as the column is, so column is now incremented to 2 3 4 how long will it come for the first row it will come 5 times because rows minus rows 5 minus 0. So, I will print this and then I will come back this loop is over, this loop is over, I will a print a backslash n up to this sorry I am sorry this for loop is actually extending up to this. So, from here I go back. So, I come to the second row.

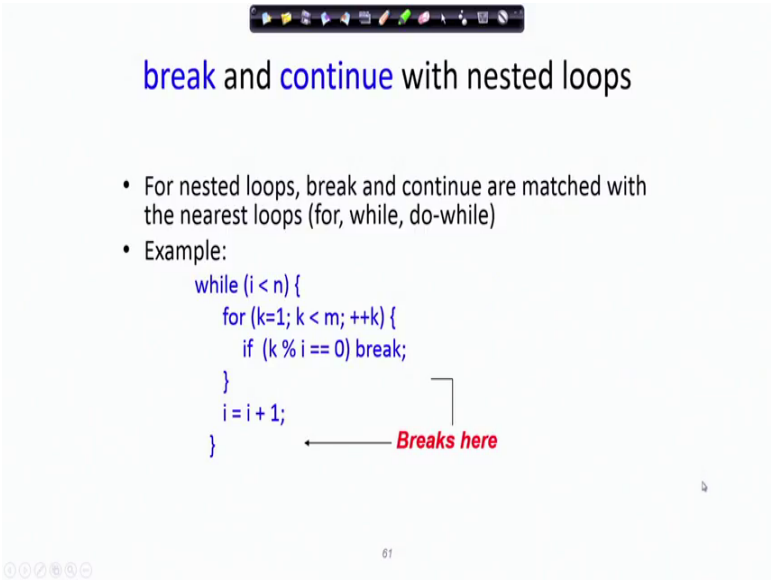
(Refer Slide Time: 26:37)



Now, row becomes 1 let us see, now row becomes 1. Less than 5 for column equal to 1, column less than equal to row 1 still valid I give one blank here, one blank here for the second row. My row pointer as come here, row is 1. Then I will do column 1, 2 column minus rows ones blank I have already given. So, I am. Now, my starting is here how many? Now, row is 1, 5 minus 1 so that means 4, 4 times this will loop and print star and then printf n.

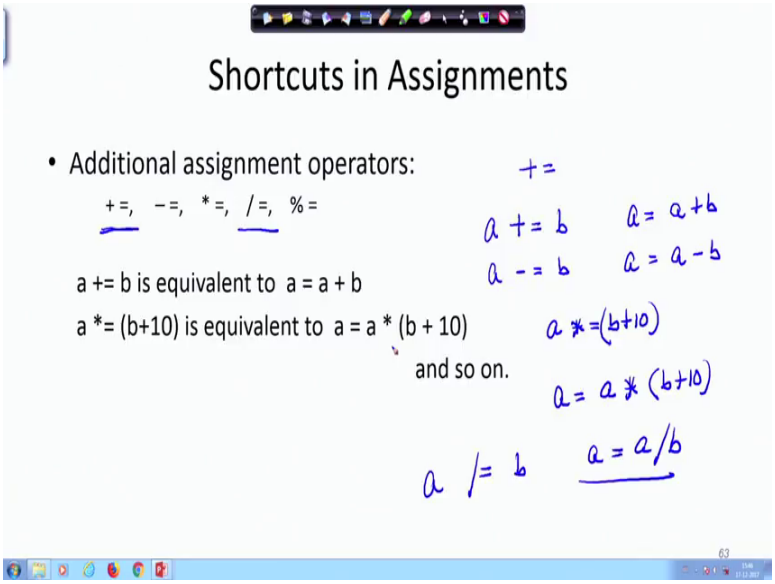
Next time I go back here and this becomes 2. So, row 2 to less than 5 column is again now 2 blanks, column is 1 2 less than equal to row and row is 2, so 1 to 2. So, there will be 2 blanks here one blank here, one blank here. So, I am coming here and then I am printing this is 2, so 5 minus 2 3 stars. So, in this way I can go on and print this figure by an intelligent way of applying the for loops or the nested loops and putting in the spaces together spaces properly. I think this gives a very interesting example for you to look at.

(Refer Slide Time: 28:23)



So, these are some of the examples that we have seen. We will come to this for thing again later, but let us just remind you a little bit about some things that we had mentioned in passing.

(Refer Slide Time: 28:31)



For example, this operator plus equal to as for example, here a plus equal to b this means a assigned a plus b these are some shortcuts all right, a minus equal to b that means, a is assigned a minus b. Here a star b plus 10, a as star assigned b plus 10; that means, a will be assigned a times b plus 10. So, in that way we have got this one also, say a assigned b

that means, a is assigned a divided by b these are some of the shortcuts. So, this is just to wrap up some of the assignment operations that we are talked about. But these are as I said that you can keep this for later use. Right now more fundamental thing that you need to know is a use of if else for, while, do while etcetera.

In the next lecture we will start with a new concept called arrays and there you will find that these loops are becoming so important and will have many interesting applications using arrays that will be done from next lecture onwards.

Thank you.