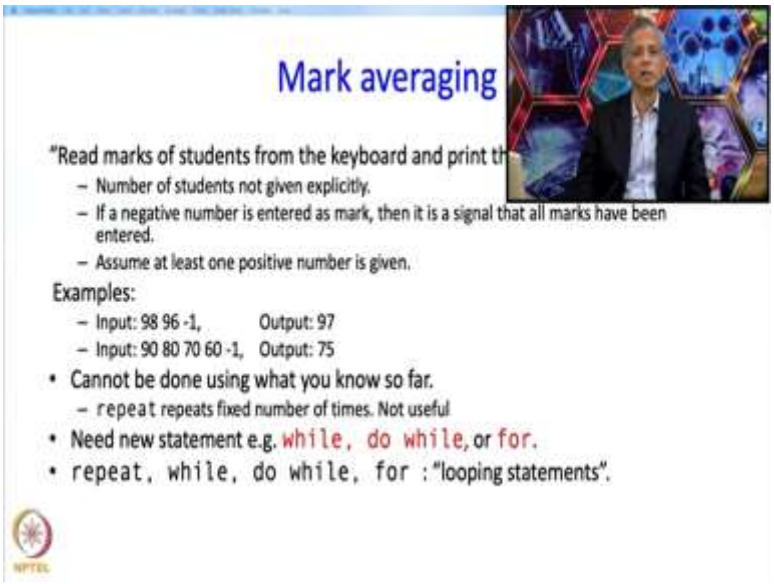
An Introduction to Programming through C++
Professor Abhiram G. Ranade
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Lecture No. 7 Part – 1
Looping Statements
Loops

Welcome to the second lecture sequence of third week of the NPTEL course on an introduction to programming through C++. I am Abhiram Ranade and the reading for this lecture sequence is the chapter 7 of the textbook.

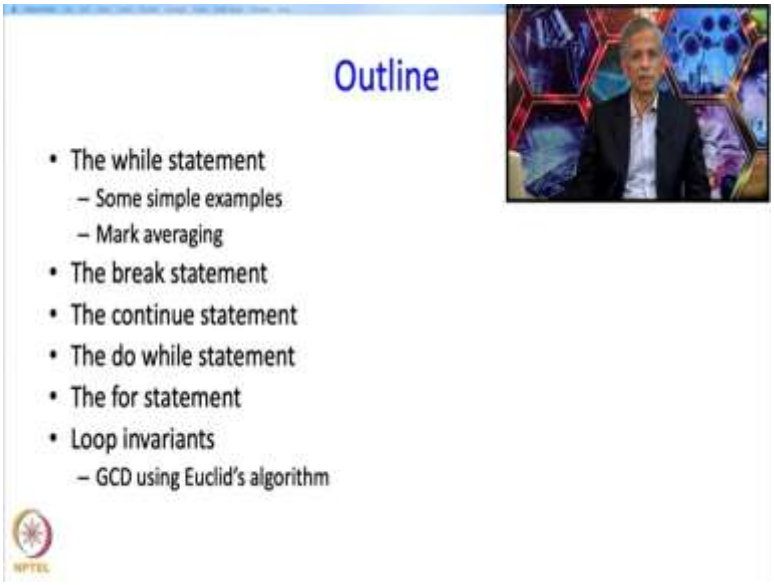
(Refer Slide Time: 00:41)



So let me begin with the problem the problem is that of averaging marks. So, you have to read your program is to read the marks of the students typed by the user from the keyboard and then print out their average. Now there are some conditions. So, the number of students are not given explicitly, instead if a negative number is entered as mark then it is signal that all marks have been entered and you may assume that at least one positive number is given. If no numbers are given, then the average is undefined and therefore, you have to assume that at least one positive number is given. So let me first construct some test cases or examples of input output. So, for example the input might be 98, 96, -1.

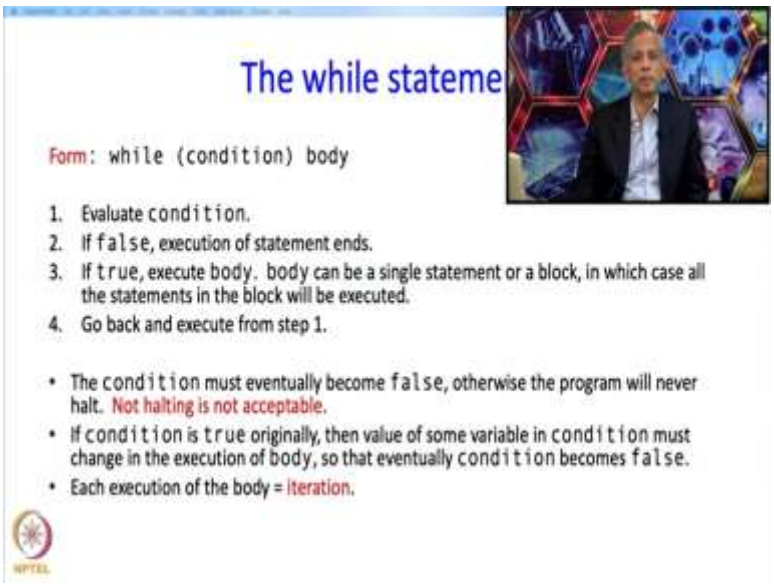
So, this indicates that the first two numbers 98 and 96 are student marks and -1 is the indication that no more input is to be given. So in that case there are two marks 98 96 so their average is 97. Another example 90, 80, 70, 60, -1. So again the actually student marks in this case is 90, 80, 70, 60 and clearly the average is 75 which is supposed to be output. Now, this cannot be done using what you know so far. So far you only know one single repetition statement which is repeat, and that statement repeats a fixed number of times. So it is not useful here because here you are going to repeat as many times as there are students but how many students there are is not given to you beforehand. So, we need something new and there are a number of C++ statements that can do what is needed over here so these statements are the while statement, the do-while statement and the for statement and the statements as well as the repeat statement are sometimes called looping statements. So, loop is just another, another term for repetition. So, loop is doing something in a circular manner so repetition basically.

(Refer Slide Time: 02:59)



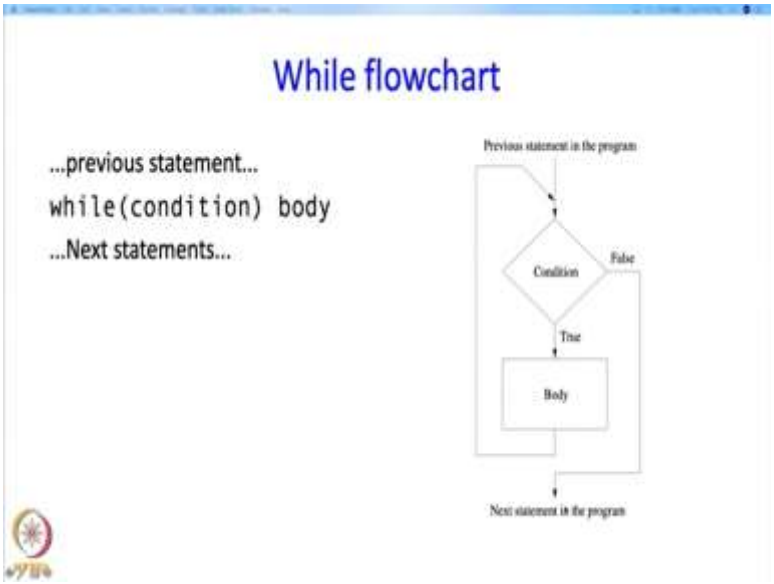
So, the outline is this lecture sequence is I am going to first discuss the while statement, I will talk about a simple example and then I will get to the mark averaging problem. Then I will talk about the break statement, then a statement called a continue statement. Then, I will mention the do-while statement but I am not going to discuss it in great detail. It is described in the book. Then I will talk about the for statement and then I will talk about loop invariants and this going to be discussed with the Euclid's algorithm for the greatest common divisor as an example.

(Refer Slide Time: 03:47)



So, let us look at the while statement. The form of the while statement is while, then in parenthesis: a condition, and then the body. So, the way this works first we are going to first evaluate the condition, the condition is simply something which evaluates to true and false ok? So we studied this last time, then if the condition is false then the execution of the while statement ends. If the condition is true, then the body is executed, body can now be a single statement or a block anything is possible, but if it is a block, then all the statements in the body will be executed. After that, after the execution of the body the execution again resumes from step 1. So again the condition is evaluated. Again if the condition is false, then the statement ends. So the statement only ends when the condition turns out to be false. So otherwise until the condition turns out to be false the body is executed as many times as needed for the condition to become false. Now the condition should become false in a well written program, otherwise the program is not going to halt. And not halting is not an acceptable outcome we want our programs to produce an answer and stop execution as quickly as possible or certainly in finite time. So, this means that if the condition is true originally, then the value of some variable which appears in the condition must change during the execution of the body and only in this case can eventually condition become false. Just like what we said for repeats, each execution of the body is also called an iteration.

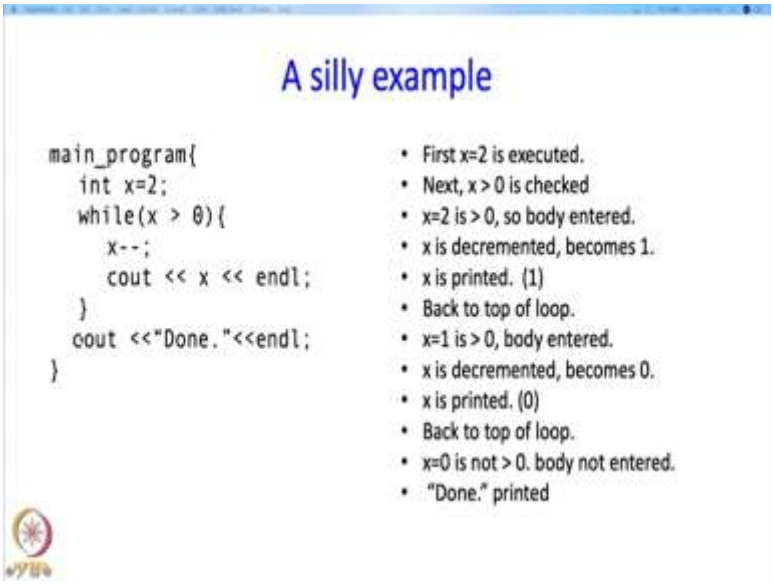
(Refer Slide Time: 05:22)



So, the while turns out to be a slightly complicated statement so it is nice to look at its flow chart, so in the left we have the while statement as a part of hypothetical program so there is the previous statement, then there is the while statement and then there is the next statement. So how is this going to execute? So that is given here on the right. So, the previous statement in the program is executed after that the condition stated in the while is executed so this statement over here this this condition over here.

If this condition is false then you go off and execute the next statement over here. If the condition is true however, you execute the body as many statements there might be in the body you execute them one after another and after that you again go back and evaluate the condition and you keep on doing this until eventually at some point the condition is false, in which case you can go on to the next statement. So that is how a while executes.

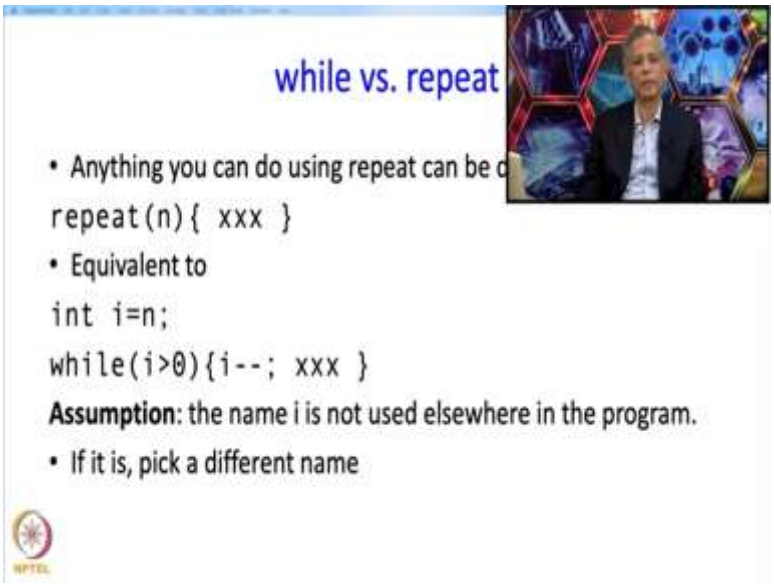
(Refer Slide Time: 07:04)



Let me begin with a silly example, so here is a main program in which we have x is equal to 2 to begin with and then there is a while loop and after the while loop a message is printed. So, let us see how this executes. So, first as the code executes from the top this statement will get executed, so x will be set to 2. After that we look at the body, so we encounter the while statement and as you might remember we have to check the body first so you check is x bigger than 0? So since x is 2 it is bigger than 0 and therefore a body is going to be entered so when the body is entered the first statement in the body is x-- or you want x to be decremented. So this means x will be decremented and the value will become 1, after that you have a cout statement or the current value of x is printed the current value of x is 1 so that will get printed. So 1 is being printed over here. Now, that ends one iteration of the while loop or one execution of the entire body. But at this point the statement does not end ok? So at this point we are going to go back of the top of the loop and we are going to recheck the condition. So, at this point x has the value 1, because we decremented it in the first iteration x has the value 1, so 1 is still bigger than 0 so the body is going to be entered. Now again there is a decrement statement so x will be decremented, and its value will become 0. So after that there is the print statement so the current value affects which is 0 is going to get printed so that is what is happened over here ok? So in this statement itself x has become 0 and the while condition seems to say do this only while x is greater than 0, but it does not mean that we are checking before every statement, we are only checking at the beginning. So, even though x is not actually greater than 0 we will still execute this statement ok? And 0

will be printed. After that we are going to go back at the top that is what while tells us to do and again we are going to check the condition. So this time however x is 0 and 0 is not bigger than 0 and therefore the body is not entered and in fact at this point the execution of the while ends and you move on to the statement following which is this print statement. So as a result of which done is going to be printed. So this was a silly example but it tells us it shows exactly how a while statement executes. Now it is good idea to think about the comparison between the while and the repeat.

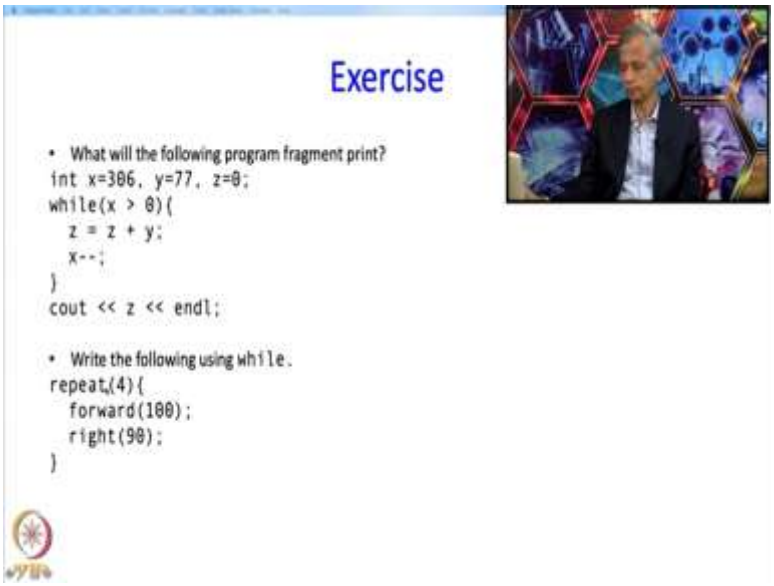
(Refer Slide Time: 10:22)



So let me first observe that anything you can do using repeat can be done using while. Why is that? So suppose, you have this repeat statement. So n is some value xxx are some group of statements, so I will show you how you will write a program, how you will translate a program which contains this statement and replace it by an equivalent while statement. So here is the translation so instead of that single repeat statement, you will have these two statements. So first you will say you will have a variable i which you are going to assign to n so that the same value whatever many times you wanted to repeat is now going to go into i. Now you are going to check is i greater than 0 so while i is greater than 0 you are going to execute xxx but you are going to decrement the value of i in each iteration. So if, so whatever, whatever value this had so whatever integer value if had say it was 10 then this loop will be executed 10 times because you would have to decrement this i 10 times only after which I would become 0. And at that at that point this condition would fail and then you would go on to the next statement.

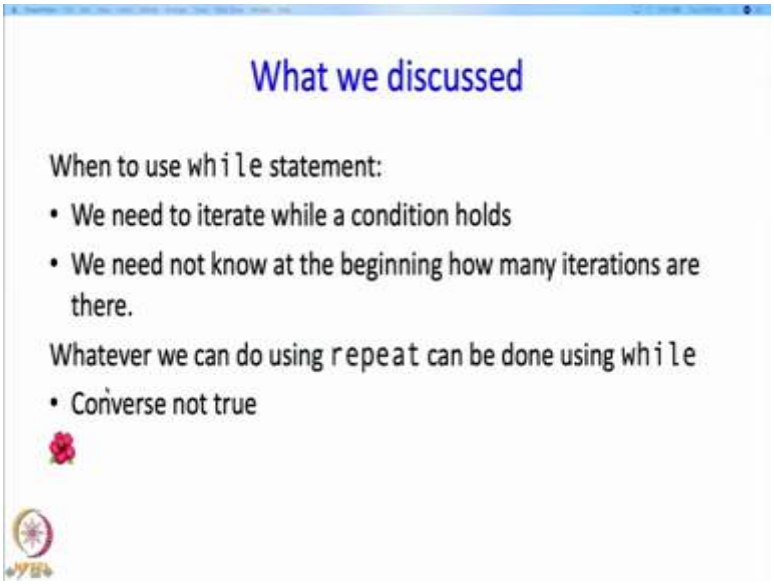
Now this, this we created a new name just for doing this translation. But if this name i is used elsewhere in the program then this might produce an error so what I should write over here or what, what I should advice you is that pick variable name which is which does not appear in your in your program and then just use that but it must be the same variable name that you use over here as you are using over here. So if i is being used then pick a different name.

(Refer Slide Time: 12:20)



Ok, so let me give an exercise so here x is 306, y is 77, z is 0 and we are doing this loop while x is greater than 0 so we are adding to z, y and we are decrementing x so I would like you to tell me what is printed at the end. So I am not expecting you to actually execute the by hand, but rather what I am hoping you will do is you will may be do one iteration or two iterations and then you will guess the pattern and you will say that at the end this number or this expression where the value of this expression will be printed as said. Then I would also like you to translate this familiar code fragment which uses repeat into an equivalent code fragment which used while loop.

(Refer Slide Time: 13:21)



Ok, so what did we discuss? So we looked at the while statement and we said that the while statement says the we need to iterate while a condition that we specify holds and then in comparison to repeat, the while statement does not need to know how many iterations are there before the execution starts.

So in the while statement a condition has to be specified and a iterations happen so long as that condition holds. Then we also saw that whatever we can do using repeat can be done using while and off course the converse is not true because in the repeat you need to know before hold how many times the loop body has to be iterated. So we will take a break here.