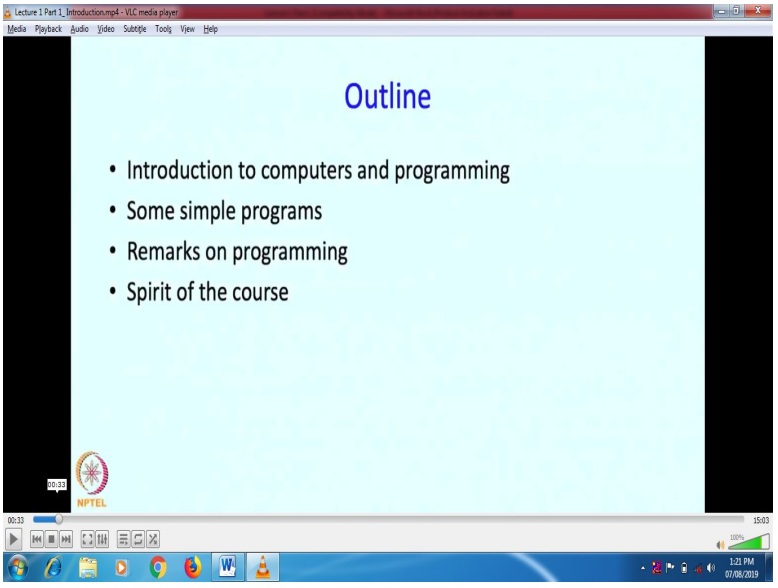
An Introduction to Programming through C++
Professor Abhiram G. Ranade
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Lecture No. 1 Part - 1
Introduction
Introduction to the course and a simple program

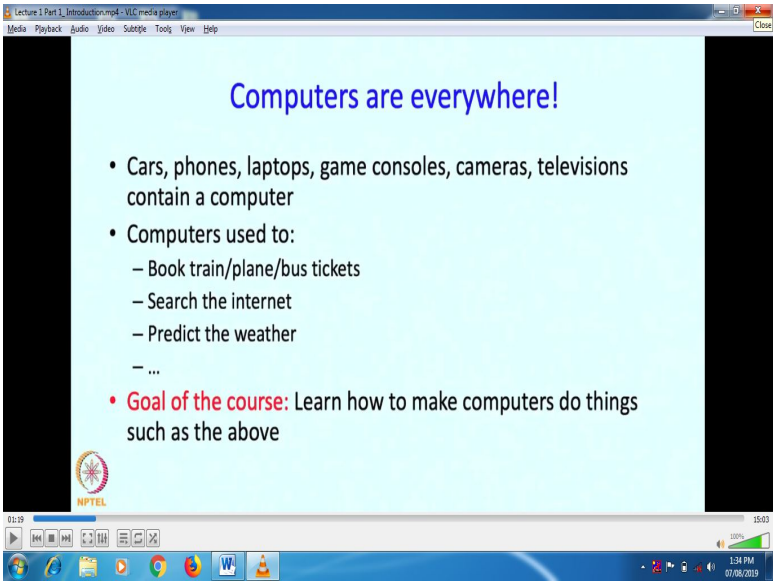
Hello and welcome to the NPTEL course on an introduction to programming through C++. I am professor Abhiram Ranade of IIT Bombay, and today's lecture will be an introduction to the course, and some material will also be covered.

(Refer Slide Time: 0:33)



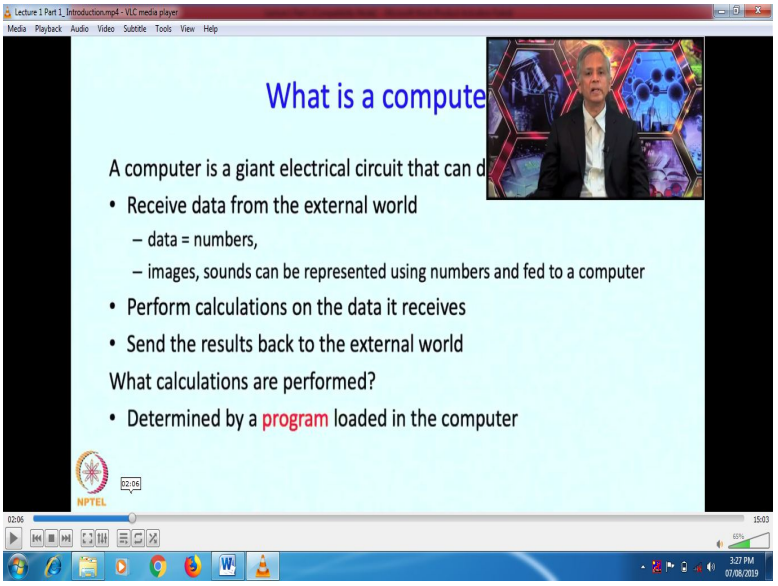
So here is what I am going to do today. I will begin with an introduction to computers and computing then I will show some simple programs. I will make some remarks on programming and then I will close with the spirit of the course.

(Refer Slide Time: 0:50)



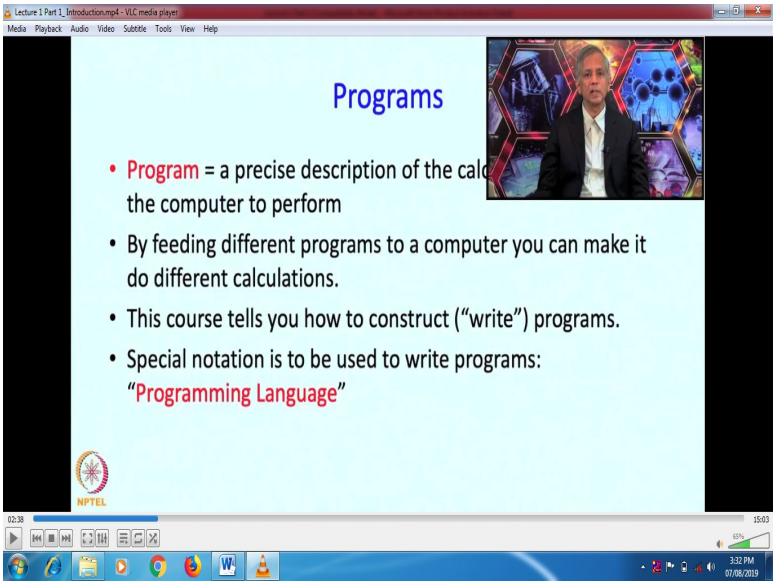
Let me begin with the observation that computers are everywhere. Cars, phones, laptops, game consoles, cameras, televisions, refrigerators, practically anything you name contains a computer. You might have used a computer to book train and plane or bus tickets, you might have used a computer to search the internet, predict the weather; maybe play games, lots of things. The goal of this course is to learn how to make computers do things such as those that we have mentioned.

(Refer Slide Time: 1:29)



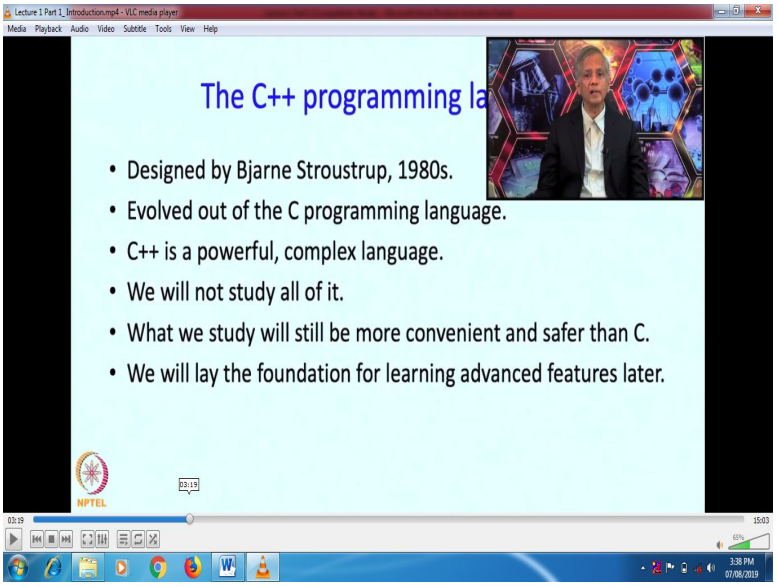
So let me begin with the question - what is a computer? A computer is an electrical circuit, it is a giant electrical circuit but nevertheless a circuit which can do the following things - It can receive data from the external world, and by data we typically mean numbers. Now it can receive images and sounds but as we will see these will be represented as numbers. A computer can perform calculations on the data that it receives, and it can send the results back to the rest of the world. Now what kind of computations does a computer perform is determined by a 'program' that has to be loaded in the computer.

(Refer Slide Time: 2:12)



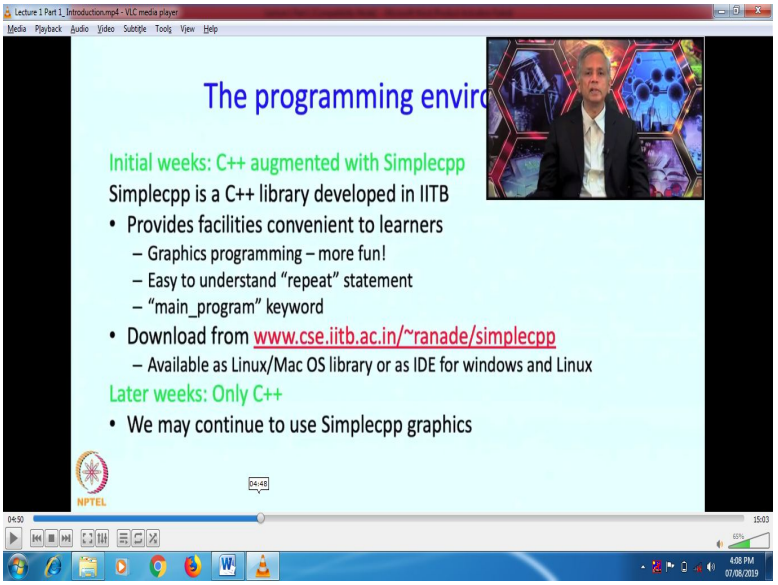
What is a program? A program really is a precise description of the calculations we want the computer to perform. By feeding different programs to a computer, you can make it do different calculations. And this course tells you how to construct programs or how to 'write' programs, which is what the process is normally called. Programs are written in a special notation called a programming language.

(Refer Slide Time: 2:43)



In this course we are going to learn the C++ programming language. This was designed by Bjarne Stroustrup in the 1980s, and it evolved out of the then-existing and still existing C programming language. C++ is a very powerful and somewhat complex language. We are not going to be studying all of it. We will study a subset of it which is still going to be much more convenient, and in fact safer to use than C. And we will lay the foundations of learning advanced features for later courses.

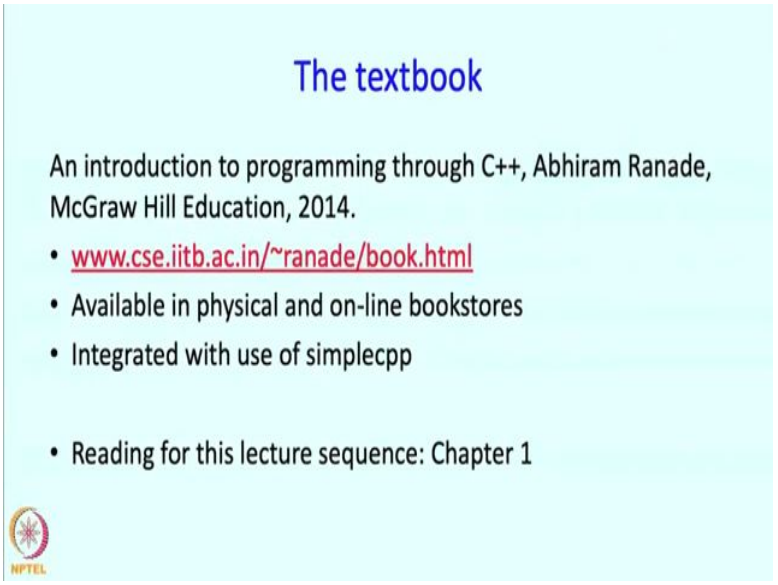
(Refer Slide Time: 3:24)



In this course, in the initial weeks the programming environment will be C++ augmented with simple cpp, where, simple cpp is a C++ library developed in IIT Bombay. Simple cpp provides facilities which are convenient for learners. For example, it allows you to do graphics; it allows you to draw pictures. This is certainly going to be more fun and probably learners, or beginners will appreciate it. Then, it provides an easy to understand statement called the 'repeat' statement and we will see that today itself.

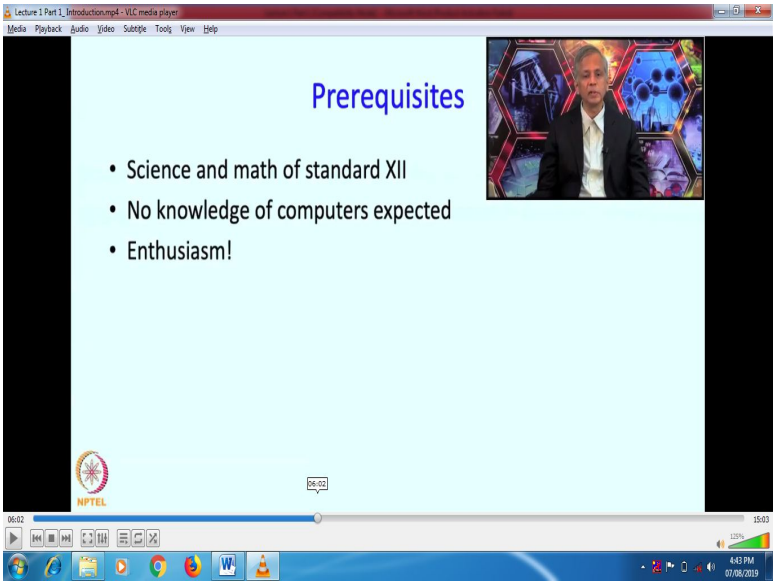
It will also provide a "main program" keyword which also we will see today. Simple cpp can be downloaded from this URL shown here: www.cse.iitb.ac.in/~ranade/simplecpp. It is available on Linux and Mac OS as a library, or as an IDE for Windows and Linux. Later weeks of the course we will just use C++; we may not use the features of simple cpp. But, on the other hand if you want to do graphics the features of simple cpp will definitely come in handy.

(Refer Slide Time: 05:01)



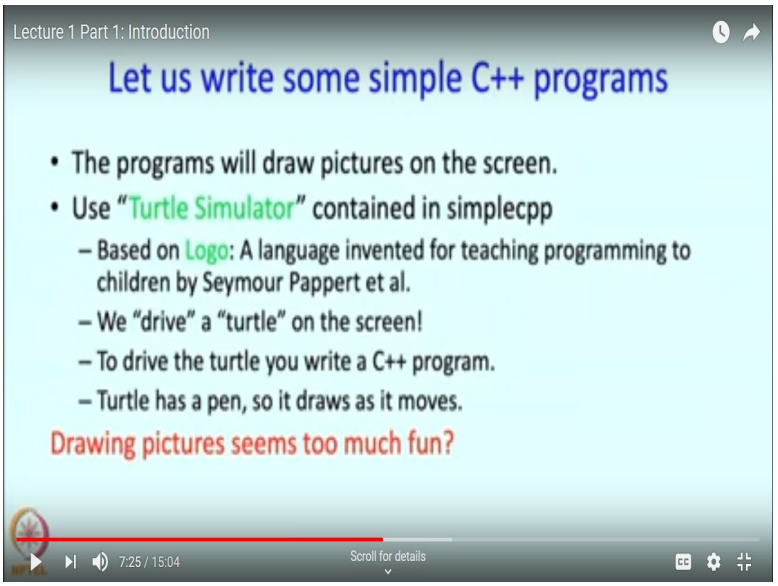
We are going to be using the following textbook: the title is “An Introduction to Programming through C++”, written by me, published by McGraw Hill Education in 2014. Here is the web page for the book: www.cse.iitb.ac.in/~ranade/book.html. It is available in physical and online bookstores, and it is integrated with the use of simple cpp. Today's lecture is based on chapter 1 of the book and you are recommended to read that chapter.

(Refer Slide Time: 5:44)



This course does not have many prerequisites. Well, you do need to know the science and math of standard 11^{th} and 12^{th}, because we will be using examples from that science and math. No knowledge of computers is expected, you will learn all of that in this course. In addition to lectures, we will have instructions and maybe even videos talking about use of computers. And more than anything you need enthusiasm. You should want to do things with computers, you should want to have fun with computers, that is really most important.

(Refer Slide Time: 6:10)

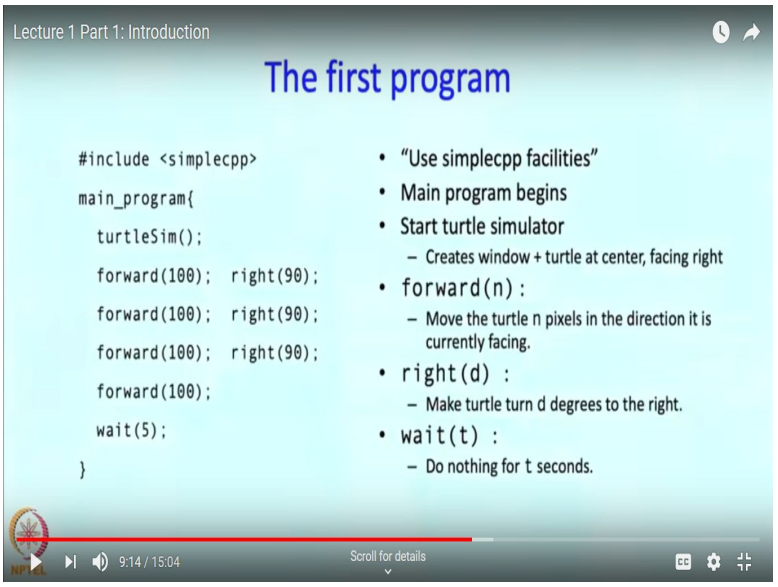


So, let us get on with the main business. We are going to write some very simple C++ programs. These programs are going to draw pictures on the screen, and they will use a so called ‘Turtle Simulator’, which is contained in simple cpp. The ‘Turtle Simulator’ is based on ‘Logo’, which is a language invented for teaching programming to children by Seymour Pappert and others in the late 1960s. It is pretty old, but you will see it is a lot of fun and later on in the course, you will realize that it is really an interesting and a useful set of tools.

The point of logo programming and the turtle simulator is to ‘drive’ a ‘turtle’ on the screen. So, you will see a small triangle typically on the screen and you are going to drive it and the way you are going to drive it is you are going to write a C++ program. The C++ program will tell the turtle what to do.

Now the turtle has a pen, so as it moves, it will draw. So that is how you will be able to make interesting drawings. Now you might think, are we learning the serious subject of programming, or are we learning drawing pictures, which seems to be too much fun? But, you will soon see, that if you master picture drawing, you will actually be mastering programming.

(Refer Slide Time 7:45)



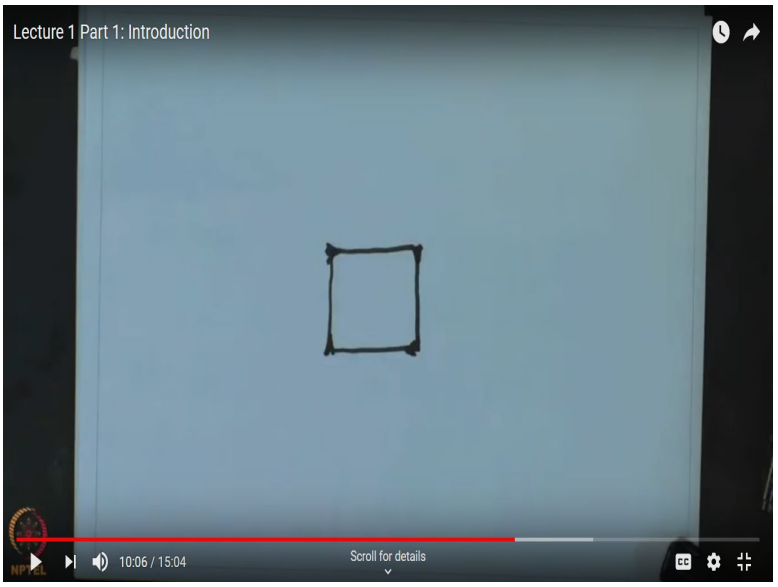
Alright, so here is the first program. So I have shown it over here. I am going to explain it one statement at a time. So the first statement 'include<simplecpp>' in those funny-looking brackets simply tells the computer, "Look, I am going to use the simple cpp facilities". Then, the 'main_program' is a keyword which says that look, what follows is the main program, so starting from the open brace all the way till the closed brace at the bottom of the page.

Then the 'turtleSim()' command starts the turtle simulator. What this does is that it is going to create a window, it will have the turtle at the centre, facing right.

Then, you see the command 'forward(100)', well in general this command is forward(n), where 'n' can be any number. So in this case the turtle is being commanded to move 'n' pixels in the direction in which it is currently facing.

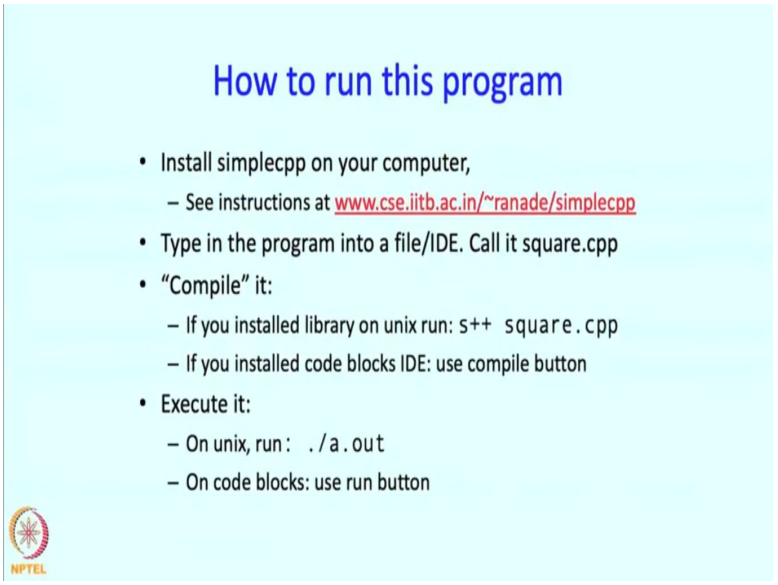
'right(D), where 'D' is expected to be the angle in degrees tells the turtle to turn right. You can have a similar left command as well. And 'wait(t)' tells the turtle to do nothing for 't' seconds, so these t seconds or in this case 5 seconds are what you are given to admire the drawing that the turtle has drawn.

(Refer Slide Time 10:03)



Now, let us see what drawing the turtle will actually do by looking at the program. So, this will start the turtle simulator and it will create a window, then the turtle will move forward by 100. So the turtle moves forward 100. So if the turtle is over here and facing in this direction it will move forward by 100 steps, then it will turn right 90 degrees, so then it will start facing in this direction, it will then move forward 100 pixels. It will again turn right 90, then again move forward 100 steps, then again turn right 90, and then again move forward 100 steps. So what has the turtle drawn as the result of this? It has drawn a square of side length 100 pixels. And after that that turtle is going to wait, and then the whole window will vanish, and the program will come to a halt. So this is what the program is supposed to have done. And let us see now how do we run this program?

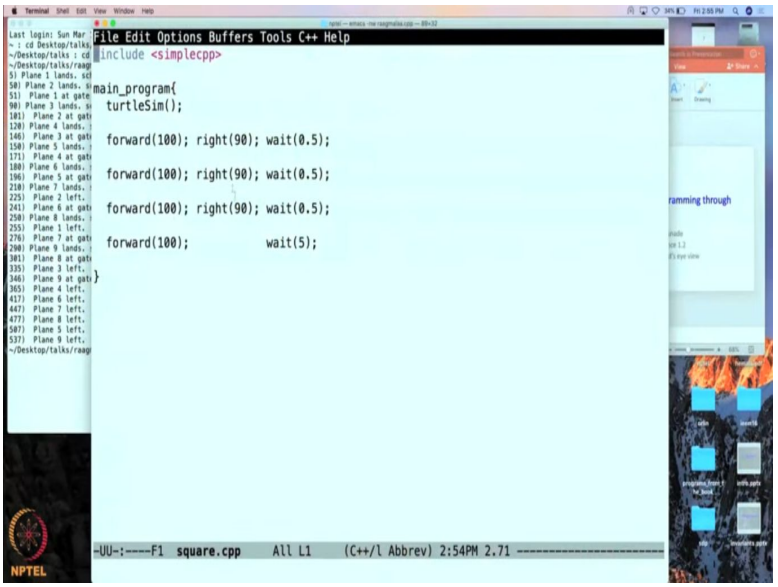
(Refer Slide Time 10:27)



So, for that purpose we need to install simple cpp on your computer. How do you do this? Well you have to see the instructions at this webpage. Then you have to type in the program

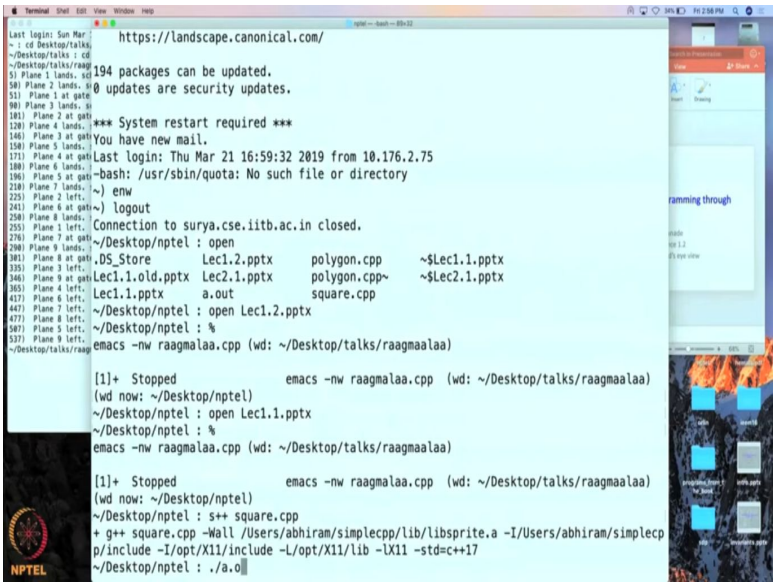
into a file or the IDE, whatever you have downloaded, and let us say you call it square dot cpp then you have to compile it. So compilation can happen by typing s++ square dot cpp if you installed a library on UNIX. If you installed the code blocks IDE, then you simply have to press the compile button. Then you have to execute it. So on UNIX you have to type 'dot slash a dot out', which is the result of that compilation process. So we will explain to you what compilation means in a little bit. But, the result of the compilation process is a file called 'a dot out', and you just have to execute it. On code blocks you just need to use the Run button.

(Refer Slide Time 11:41)



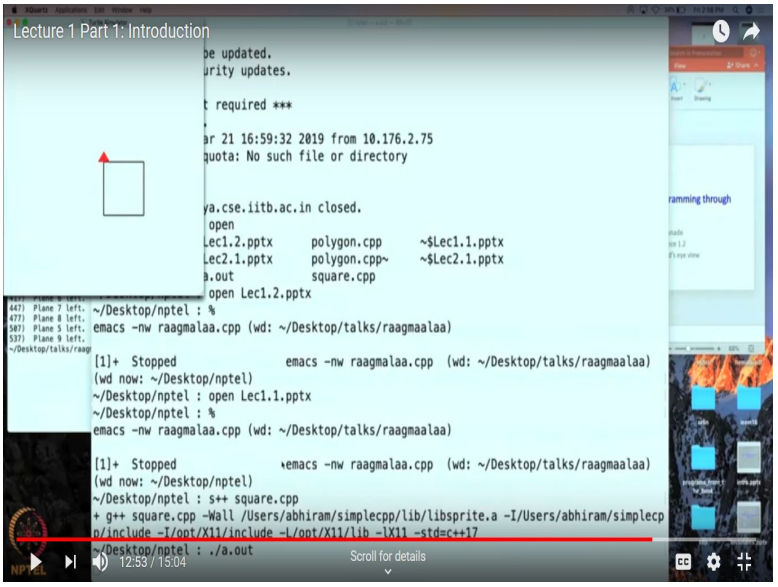
So now, I am going to show you exactly how this happens and I am going to use the simple cpp library, and I will compile the program for you. So here is the program that I showed you earlier. This program is slightly different. So you had seen that there was a wait 5 seconds at the end of that program, but here I have also put in these additional waits. So these waits - 0.5, 0.5 are going to tell the turtle to wait for about half a second after each forward and right step. If I do not do that, then a computer works very fast, so the turtle will move very fast and before you see it, everything will be drawn. And you will not really be able to see the movement in any nice way. So therefore, we have put in these waits in these additional waits. So this has been typed into an editor and it is now in a file called 'square dot CPP', the name of the file is appearing over here as far as the editor is concerned, but you might you might have it in your IDE.

(Refer Slide Time: 12:32)



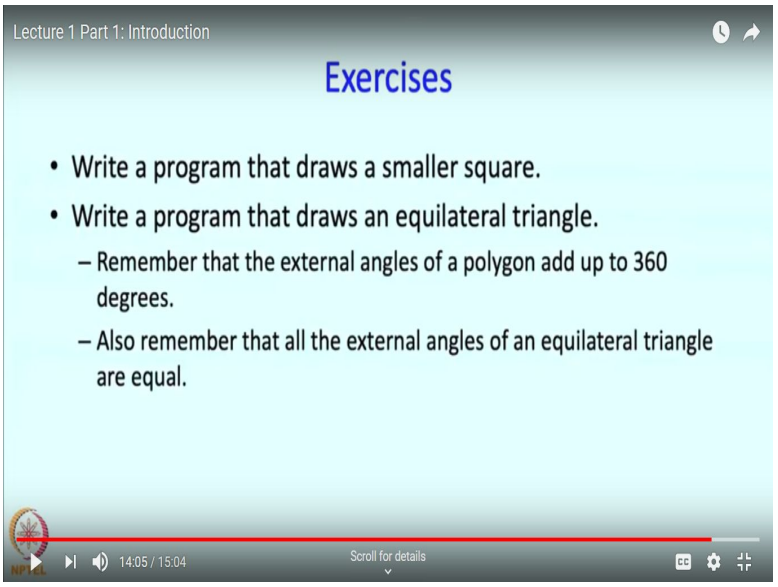
So, let me now try to compile that file so for this I am going to say s++ square dot CPP. So this will compile the file. And now I am going to execute it.

(Refer Slide Time: 12:50)



So as you can see, the red triangle appeared it drew a square and now it is gone. So this is what that program did. And this is what you can do, you can change the program, you can draw other things as well, as we will see soon. So you saw that program execute.

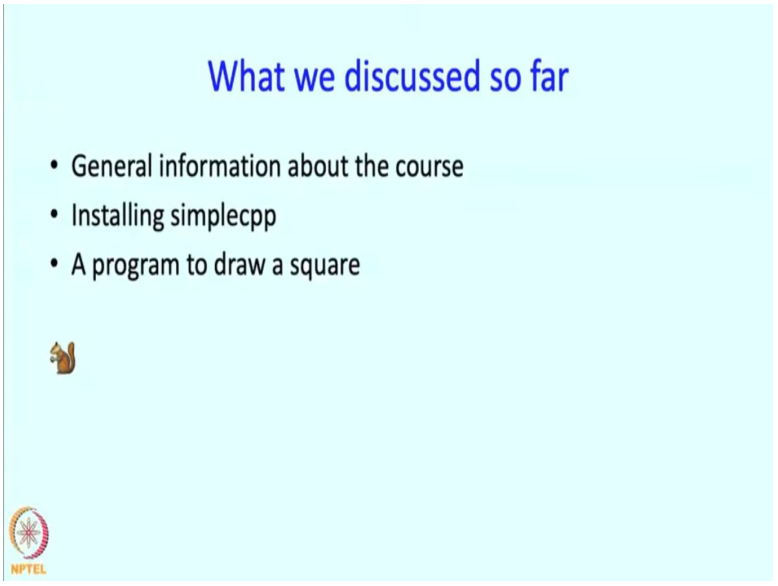
(Refer Slide Time 13:46)



Now, I will suggest that you become familiar with that program and maybe you change that program a little bit. Do not make many major any major changes, but see if you can change it so that it draws maybe a square which is 50 pixels on the side. Basically we just have to change that 100 to 50. But do it, so that you get confidence of running something on a computer. Likewise, a slightly bigger change is to make it draw an equilateral triangle. Well, for an equilateral triangle, you will just have to draw 3 lines instead of 4, and furthermore, the angles will have to be different.

For this purpose, remember that the external angles of a polygon add up to 360 degrees. And, if the polygon is a triangle, then there are only 3 angles, and all of those exterior angles are equal. So therefore each angle must be 120 and that should be the turning angle.

(Refer Slide Time 14: 25)



So, what have we discussed so far? We have discussed general information about the course, we have talked about how to install simple cpp and, we have talked about a program to draw a square. So we will take a break and resume in a bit.