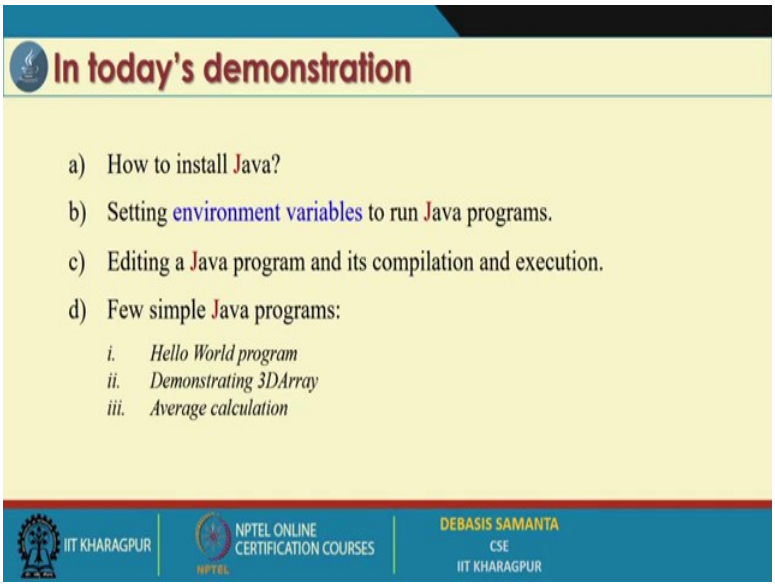
Programming in Java
Prof. Debasis Samanta
Department of Computer Science Engineering
Indian Institute of Technology, Kharagpur

Lecture – 04
Demonstration – 1

So, we have learned about how to run Java programs and then the different resources which are required to support your Java programming. So, today we have a quick demo about setting our environment suitable for Java programming. We in addition to this we shall discuss a few more things also.

(Refer Slide Time: 00:40)

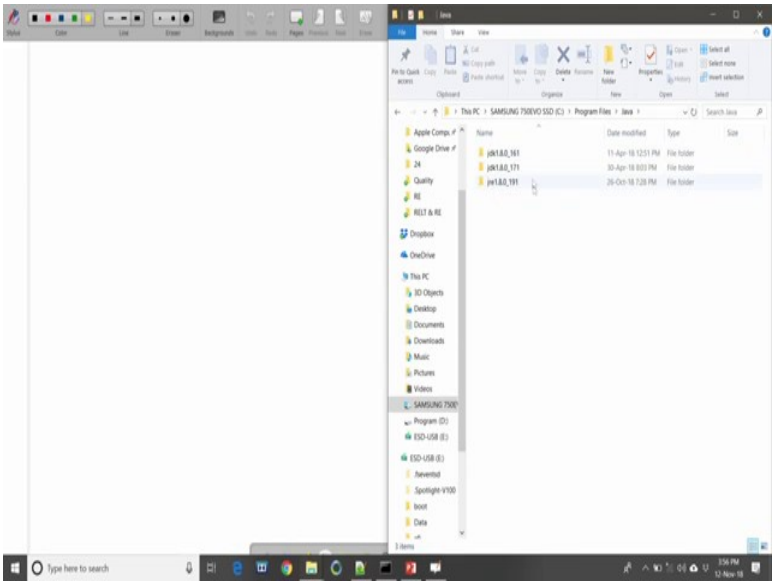


So, you should configure your machine right if you use a laptop or any PC so, that you can run your programs from your project directory. So, we should advise you to build a directory so, that you can store all the programs that you will develop. And, then we will also learn about how a program can be edited and the different steps towards the execution of your program. Finally, we shall have some simple demonstration of simple small programs so, that you can understand how to deal with a basic Java programming features.

Let us have some demo, I understand that so, yes. So, in the theoretical discussion, I have mentioned that you should install JDK version 8. I hope you have already downloaded

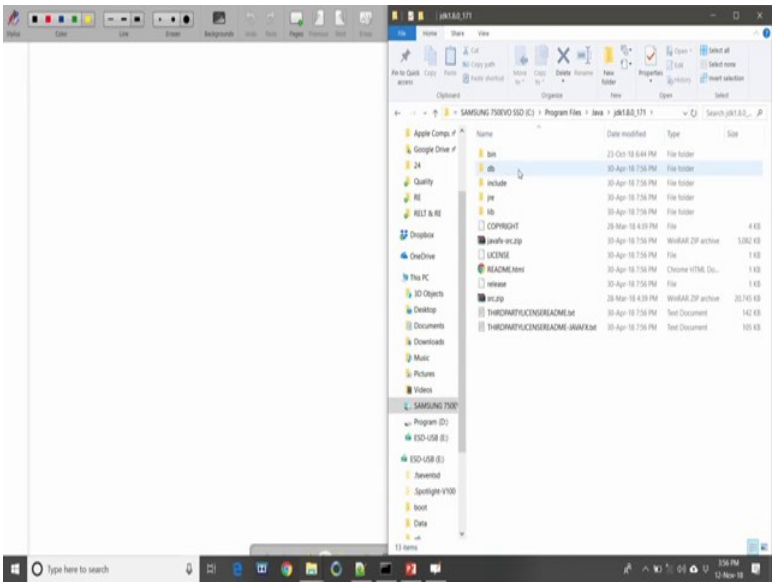
the JDK 8 and installed in your system. Now, suppose you have installed your JDK software in program files.

(Refer Slide Time: 02:21)



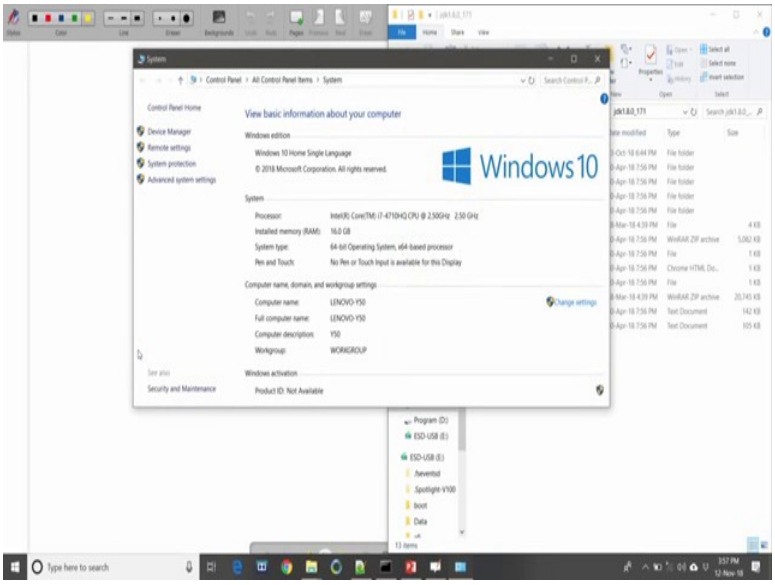
So, we will see exactly we have installed in our own machine in program files directory the JDK system, you can see we have installed our programs I mean JDK in program files directory; the name of the directory here is JDK. And, this is the version where we have installed the JDK version 8 ok.

(Refer Slide Time: 02:39)

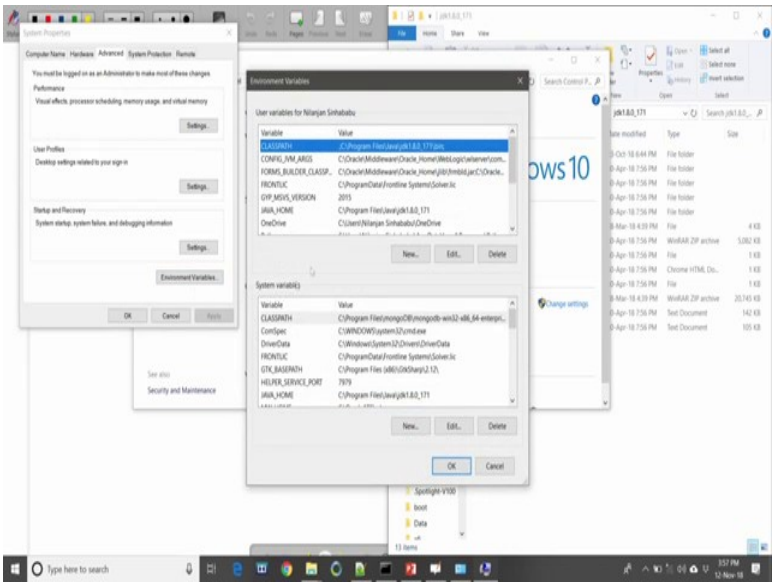


Once the installation is successful you have to set the path of your program directory; that means, the main directory where the all the Java tools, Java codes, Java commands are installed. So, we have to do it I am giving you the step by step.

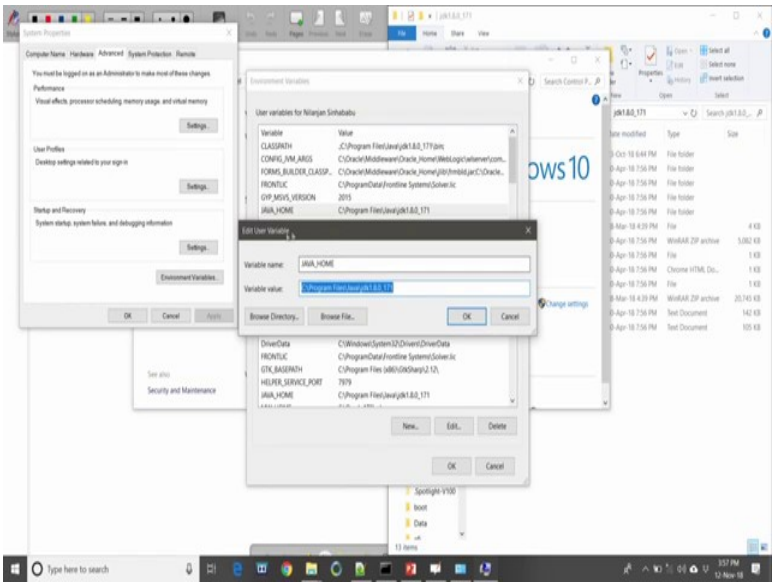
(Refer Slide Time: 03:04)



(Refer Slide Time: 03:10)

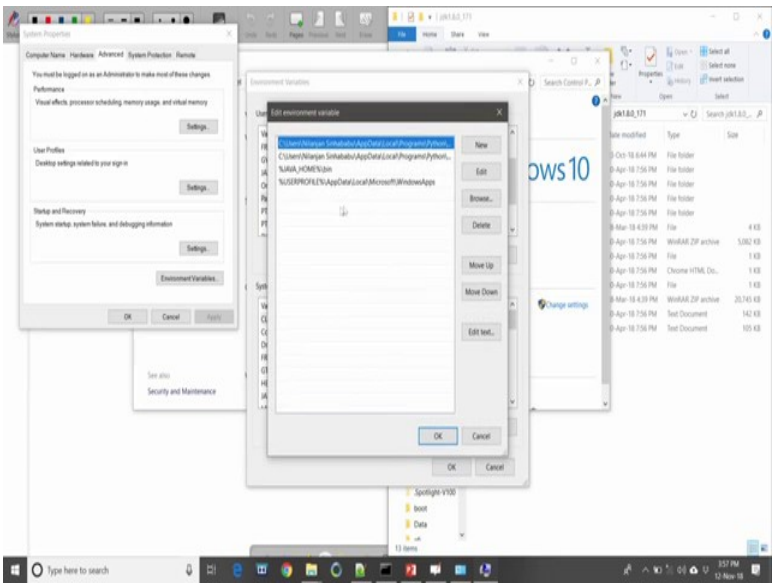


(Refer Slide Time: 03:14)



First you have to go to properties and then you go to advanced settings and the environment variables and then you have to set the Java home, the variable name JavaHome then the variable value for the Java home should be set at the C://Program Files/Java/JDK 1.8.0_171, this is basically the program which you have installed. So, you have to set this variable name Java home with this value.

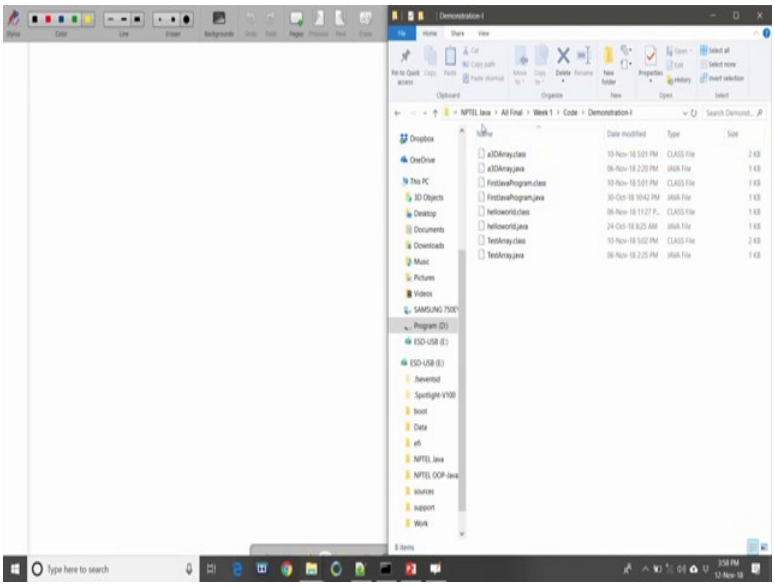
(Refer Slide Time: 03:39)



And then you have to set the path. So, the path that you should set using this name that JAVA HOME then under the JAVA HOME the bin because the bin is basically the

executable of all the Java course, Java systems is there. So, this is the setting of the parts and then finally, you have to set the classpath. So, you can set the classpath variable by C://Program Files/Java/JDK 1.8.0_171/bin. So, this is basically your executable or the common files where it will be is it will it is stored. So, you have to set the paths: one the paths and then system variables are configured this way, it will help you to run your program from any directory.

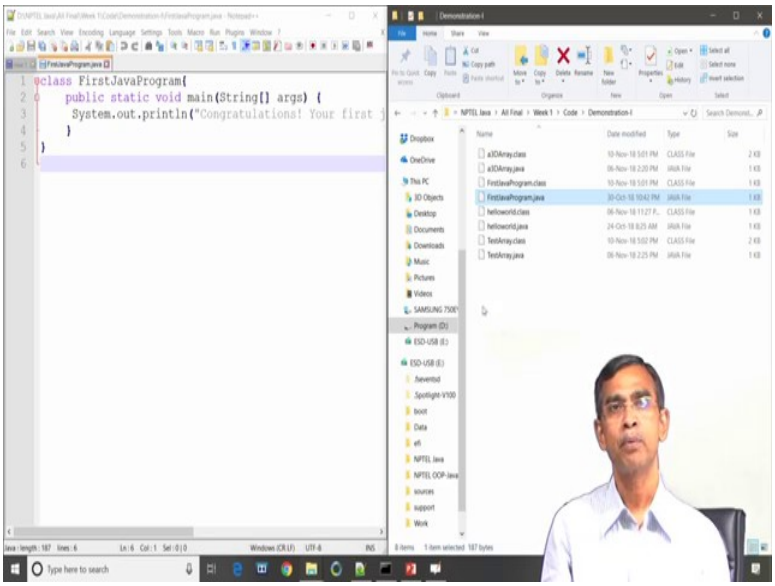
(Refer Slide Time: 04:40)



So for example, in this case, we have created one directory. The name of the directory is NPTEL Java > all final > week 1 > code> demonstration 1. So, today we have created a directory we [ca/can] can call it as a project directory, the name of the directory is demonstration 1. You can according to your own understanding you can create your project directory anywhere in your machine, in your system, in any drive. And, then once you do it you will be able to now create your program, compile your program and executive your program.

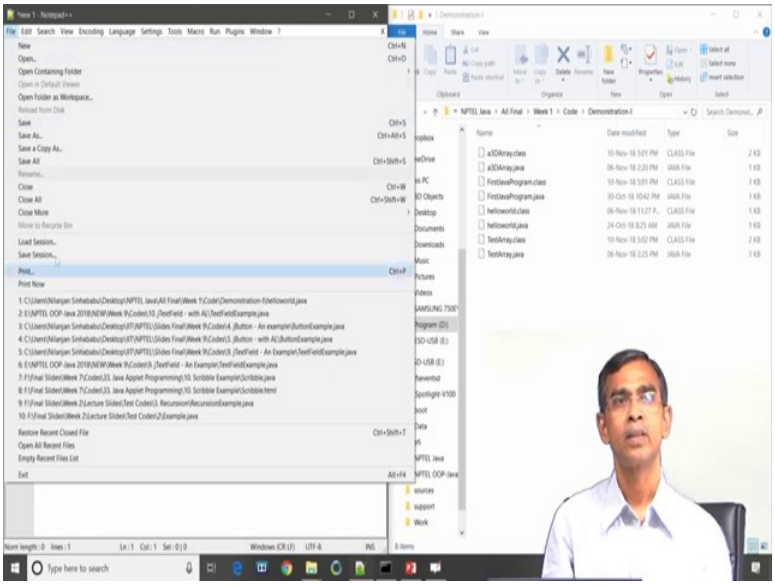
Now, I told you the Notepad++ is a very good editor which you can consider to write your program. Now, I am just opening this Notepad++ software so, that you can see.

(Refer Slide Time: 05:27)



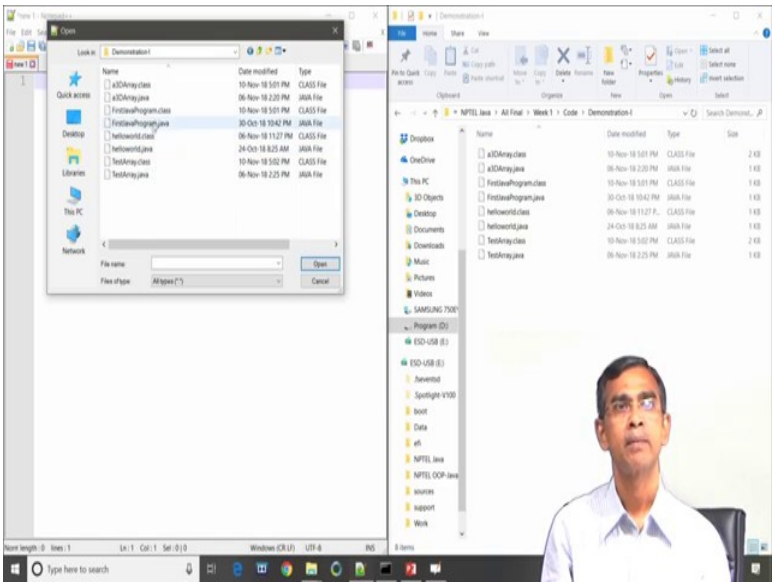
So, here is basically opening the software Notepad++ So, this is the window of the Notepad++. So, it is basically text editor you can type anything whatever the things you can type from the keyboard. So, this is the simple editor anybody can use it ok, our objective is to write a Java program; I am writing a Java program.

(Refer Slide Time: 05:53)



So, let us start Java program is like this you just ok.

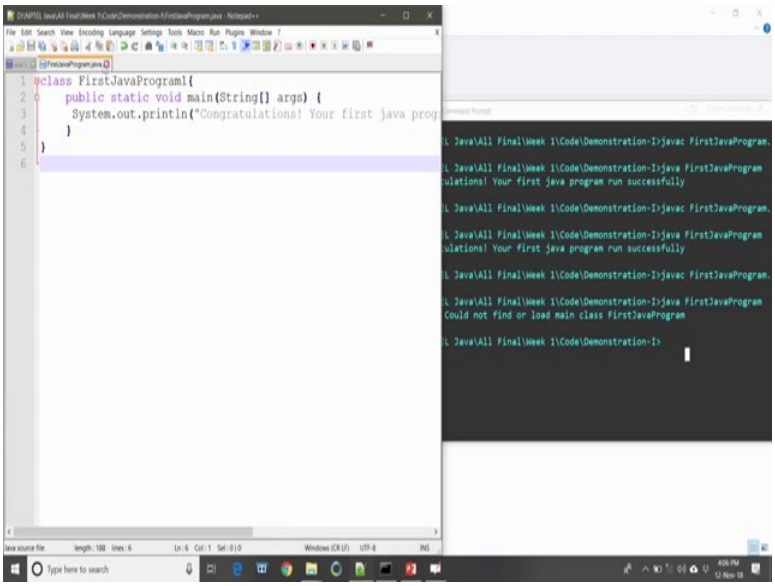
(Refer Slide Time: 05:56)



See the Java program that I am going to write it, I will not write it here through keyboard; I have already written for you I am just opening this file. So, I am opening the programs in the first Java program. So, this is the first program that we are going to have a demo. So, this is the program whose name is the, it is class name is rather FirstJavaProgram. So, this program is now supposed you have typed it from your keyboard then our next task is to save this program. So, there is a save menu and you can go save menu save as so, let the FirstJavaProgram is saved.

So, this so, the program is saved and after saving you can go to your directory where you can see the program is already saved here. For example, here FirstJava FirstProgram.java is saved here ok. Once the program is written and saved successfully our next task is to compile it.

(Refer Slide Time: 07:02)



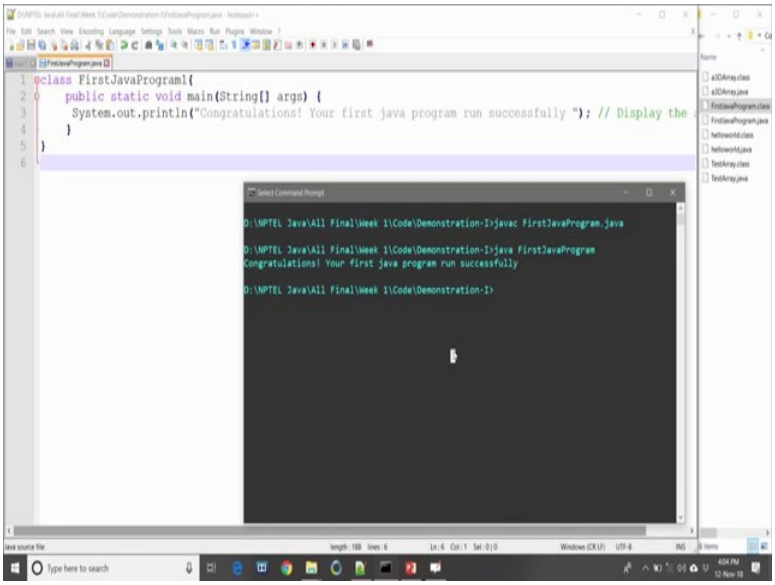
For this compiler, we have to go to the command prompt. So, you have to go to the command prompt from your Windows system if it is there and if we use the Unix system then you have to open a terminal so that you can use it to type the command. Now, in order to compile your program the command that is required is javac and then you have to give the full name of the file. The file that we have used to save our first program is FirstJavaProgram.java. So, we have to give the name of the file in the same way as the name of the class you have given and it is a case sensitive so, be careful about that.

So, this is the example where javac we have invoked the compiler to compile the program FirstJavaProgram.java. Now, let us compile it ok. So, as there is no error, no message in the command from this means that this program has successfully compiled. Once on the successful compilation, you can see in the same directory one file is created, the name of the file is the same as the name of the Java file except the extension is .class. So, here you can see the byte code file which has been created is FirstJavaProgram.class.

So, one this program is successful on compilation now we are ready to run it, to run this program the command that we said to use it java. So, Java first program FirstJavaProgram and then .class you can use the .class or even if you use the class also no issue so it will run. So, here, for example, the class file name is FirstJavaProgram. So, just simply type java and the name of the class file, namely FirstJavaProgram here. So, this is the program that has been executed and as you see this program is basically used

only one statement namely System.out.println() and within this println() is basically typed the “Congratulations your first, just open or just browse.

(Refer Slide Time: 09:16)



So, this is the message that it will display “Congratulations your first Java program run successfully. Now, we can see that output also gives the same thing. This is basically the steps that you have to edit your program, compile your program using javac and then run your program using Java command. So, these are the few steps that you should consider and now say, suppose there is a mistake in this program.

For example, you have given the name of the class file FirstJavaProgram1, but you saved this program as a FirstJavaProgram.java saved this program as a FirstJavaProgram.java right. So, yes this is the program that we have created, but the see the name of the class file that we have created is FirstJavaProgram 1, but the name of the file that we have saved it FirstJavaProgram.java. The two things are not the same; this means that in this case compilation should not be successful.

Now, let us run the program the same javac FirstJavaProgram.java and then run it. No, you have to no your program, no class file is what.

Student: Different from the class file name.

So, how it is compiled successfully?

Student: Compilation will be done. The compilation will be successful.

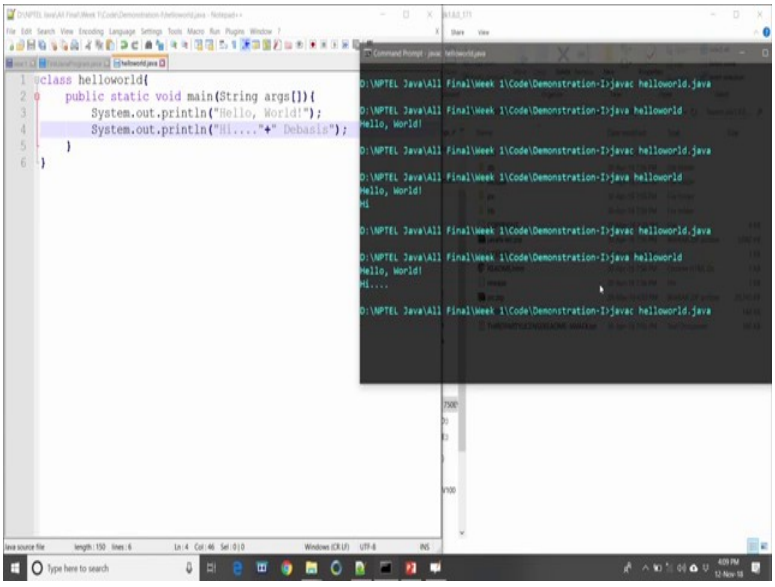
Now, let us see run the program. So, to run the program java FirstJavaProgram right now what is going on here, how the compilation will be successful. Why we have saved the program as JavaProgram1.class.

Student: Yes.

Yes, now here we can see we have saved the program file as the name FirstJavaProgram, but the name of the class that is there in the program is FirstJavaProgram1. So, there is no match, if there is no match that means mismatch then the compilation will not be successful. So, for example, in this case, we could not compile the program successfully. So, here you should note that the name of the class file should be same as the name of the Java file. Then, in that case, the program will be compiled successfully and if it is successfully compiled then you will get some execute some if the execution will be possible.

Now, let us see I have another simple demonstration of another program. So, HelloWorld yes so, this program and I am writing. So, I am writing another program. Let us give the name of the program is helloworld.java program. So, HelloWorld this is the program and it has only one statement, it is very similar to the FirstJavaProgram.

(Refer Slide Time: 12:45)

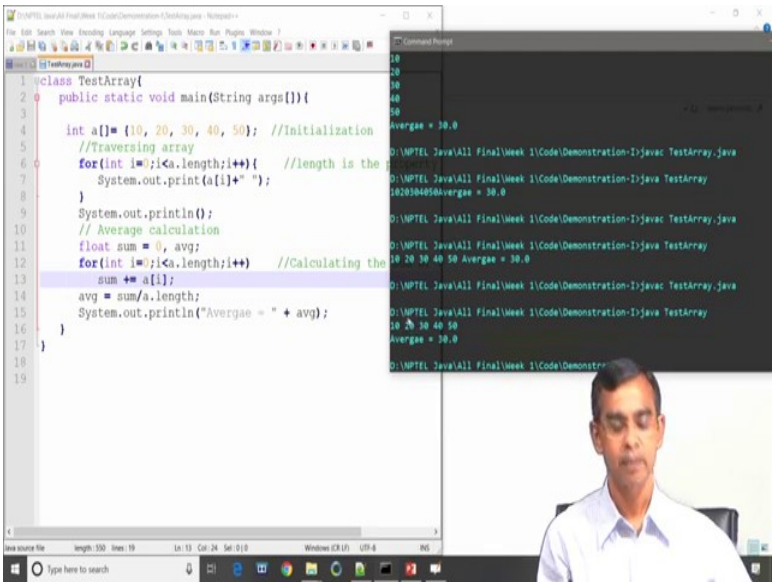


I will combine it so, javac this program has been saved as helloworld.java. So, here the name of the file Java file is helloworld, all in small letters helloworld.java. So, program compilation is successful then we will be able to run it. So, running the program using Java right so, java helloworld . right yeah; so it gives Hello, World. Now, in the same program I am adding one more statement System out . println() System.out.println(). So, here instead of "Hello, World" I am typing "Hi". So, here the two messages: the first message "Hello, World" and the second message "Hi" will be printed.

I am again compiling and then again execution. So, you see Hello, World and then Hi. So, you can understand about, if you want to print some message on the screen the statement that is required in Java program is System.out.println(). And, whatever you write within double quote is basically print will display on the screen (Refer Time: 14:18); that means, as if whatever it is written there. So, Hi for example, again Hi . . . within the double quote and the same thing Hi . . . right Hi . . . ok; now if you save it again run it so, it is like this.

Now, if you want to print two messages in one println() statement so, we usually use plus symbol so, Hi plus no no Hi ++ Debasis ok. So, now here basically in one line, it will print Hello, World then the next line Hi and then Debasis will be printed you can see ok. So, now you have hope you have understood that is a very simple statement and the simple program indeed. So, how you can run the simple program? This is a good starting point for you actually ok. Now, let us consider another program, this program is basically we have discussed how to define an array and then how to initialize an array and how to use the element which is stored in an array.

(Refer Slide Time: 15:32)



So, this is the one program that we have written here the name of the program is TestArray, where capital T and then capital you should note it. So, this is the name of the program your Java program and the and this program should be saved as TestArray.java. So, this program has been saved as a TestArray.java in the directory, the current project directory the test right. So, this is the program TestArray.java we have stored it.

Now, let us look at the program state statement by statement, we have to understand this program what we have done it now ok. So, this is the name of the class the TestArray, this is the name of the program and then public static void main String args. It is as usual there the standard syntax that you have to follow; you will understand the meaning of this statement later on.

Now, next is basically we have declared an array type of integer. So, int a within square bracket and then it is basically initialization memory allocation and then declaration all the three things have been done together. So, this is the statement int an array symbol equals then within second brackets 10 20 30 40 50 so; that means, the array a is of now size 5. And, this array contents 5 elements which are stored in a 0 location 10 a 1 location 20 a 2 location 30 a 3 location 40 and a 4 location 50. So, this array is stored and then the next one for loop, this for loop basically we will print the elements which are stored in the array. So, this loop will roll starting from i equals to 0 because the array index is from 0 and then it will loop till value of i less than a.length().

So, a.length() is basically what is the size of the array a. So, a.length() is the specific one function that you can use and that will return the size of the array. So, in that case, a.length() is 5 and then the next is basically an implementation of the loop variable i ++ and within this loop System.out.println(). So, it will print one element in the array at a time and then go to the next loop and so on. So, this will print the array. So, instead of println() if we type print it basically will print all the elements in one line. So, ln and print the difference is that is basically in one line or in a different line.

Now, let us have the ln I will discuss simple print as an alternative to println() later on. Now, once the array is array elements are printed our next step is to calculate the average value of all the elements which are stored there in the array. So, we have declared one variable called the sum and it declared as a float and then the average has also a float variable. So, we declared two variables; one is the sum and the another is average and sum is initialized a 0 at the time of declaration.

Now, so the next loop is to find the sum of all the elements and finally, it will basically calculate the average is basically sum divided by the total number of elements stored in the array that is the size of the array. And finally, there is a System.out.println() and it will print the average value with this statement that average equals the average value.

Now, let us run this program using javac TestArray.java is the program file name ok, run it run ok. So, this program is successfully compiled, now we are going to run this program right. So, here you see the first few lines 10 20 30 40 50 is an outcome of the first for loop, it basically it display all the elements which are stored there. And, the next for loop calculate the sum of all the elements the stored in the array and finally, calculate the average which is printed within the System.out.println() average equals to 30.0 in this case.

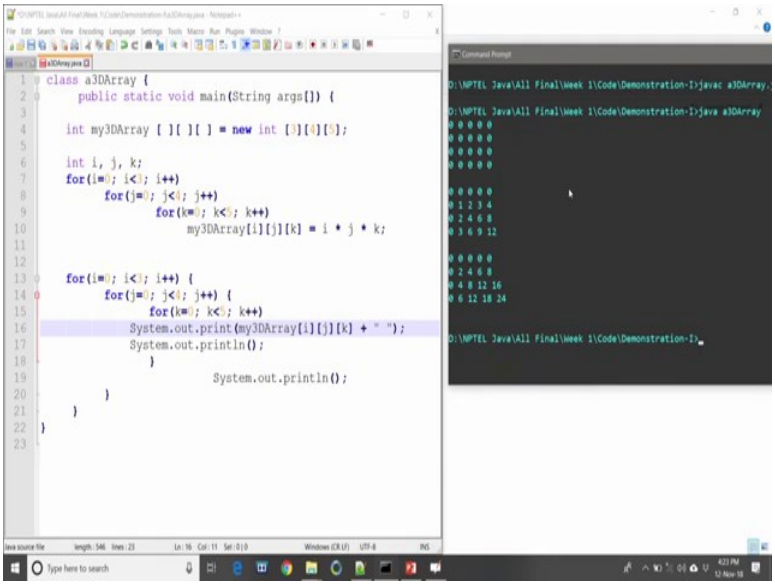
Now, I will just want to do some changes here in the first for loop, I am writing system out println() instead of println() simple print and then let us see what will happen right run. So, it is little bit not easy to difficult I can put some single space. So, I can go to the print again print a.i ++ and then within double quote plus double quote blank space. So, if we give a space so, the display will be quite comfortably visible.

So, we make the print statement like this. So now, here we can see how the output is displayable on the screen. I can make a little bit different output at the end of the right

System.out.println() at the first for loop, at the end of the first for loop brackets you know here right here, here you can type right one System.out.println(). System.out.println() just only brackets double quote right nothing ok.

I have so this semi colon so, I just see so in the first for loop it will print all the numbers in one row, then at the end of the for loop it will go to the next line and then in the next for loop it will calculate sum and then print at the next line. So now, you see the display is more comfortably presentable so, that you can see about it, yes now you can understand. So, this is the matter of simple how you can configure your printing on the display. So, using System.out.println(); regarding this System.out we will learn a lot when we will discuss this method in details.

(Refer Slide Time: 22:34)



So, let us come to another one example. In this example, we will consider here the 3DArray. Now, let us give the name of this program as a 3DArray as the file and you should save this program as a 3DArray.java. So, this program if you see the name of the program file is 3DArray here public static void main String args it is as usual ok. And, now we have declared here. So, here we declared an array of type integer and we gave the name of the array as my3DArray. As it is a 3DArray so, it should have 3 square brackets as it is shown here and then this array will be declared and then memory allocation for this memory allocation, we use the keyword new; new is basically the memory allocated.

So, it will allocate the memory for the array and here you see 3 4 5 are the 3 dimensions. So, the first dimension is for the number of pages, the second is for the number of rows and the second next last dimension 5 is the number of columns. So, if it is a 2D array then is a 4 5, if it is a 3D array then 3 4 5 like so, 3 dimensions. So, it is basically is a collection of 3 2 dimensional array 3 2 dimensional arrays where each 2-dimensional array of size 4 rows and 5 columns. So, here basically if we print it then it will print as a 3 2 dimensional arrays actually.

Anyway now, just let us see how we can initialize, there are you can read the total number of elements which can be stored in this array is 60. So, 60 array typing from the keyboard sometimes is very difficult anyway. So, we are just using one for 3 for loops to initialize this array. Now, to initialize this array I use one I mean is the nested for loop I should say, here see the nested for loop. This loop will roll for 3 loops actually the innermost loop with k = 0, k < 5, k ++.

The innermost loop is basically 2 read the elements in an in a row for each column in that row. Then the next the outer most loop the for i j equals to 0, it j less than 4, it basically we will try to read for all the rows in 1 2 dimensional array. And then finally, the outermost loop is basically read each 3-dimensional array, each 2-dimensional array in 3D arrays.

And, here the statement my3DArray i j k for any values of i j k will be stored as is a product of i star j star k star. So, this way we will we initialize the array and then finally, the next 3 loops again to print the array. Now, so the statement that we have considered to print the array is basically again 3 loops same as the loading the array, initialization of the array in the same way; only we have used that System.out.print() my3DArray the value it will print. And then finally, the last statement to give to the cursor into the next line. So, this is the program which basically declared a 3DArray initialized the elements into that array and finally, print the elements in that array. So, now let us run this program, but before going to run we should compile it so, javac right yes. So, the program compilation is successful; now run the program using java ok.

Now, here you see the output. So, I told you that 3DArray can be displayed 3DArray is basically I have to display the 3DArray in the 2D a plot. So, that is why we did it by 3 2 dimensional arrays. So, the first 2-dimensional array is for the first we can you can call it

as a phase. So, 1st phase and then the 2nd 2-dimensional array for the 2nd phase and then the 3rd 2-dimensional array for the third phase and, you can see in each phase there are how many rows.

Student: What was the.

What is the 4 cost?

Student: 5 (Refer Time: 28:02).

4 rows and 5 columns so 4 rows and 5 columns, in each 2-dimensional array there are 4 rows and 5 columns. So, this is the way we can handle the 3D array. Now, as you have already learned 3D array simply learning the 2D array also similar. So, in that case, it you just do the same program only you have to use 2 for loops for initialization and 2 for loop for printing all the elements. So, it is given as an exercise for you, you can try just repeat the same thing, declare a 2D array, initialize it using some i star j like and then display the elements ok. So, you can practice it so, that you can learn it.

So, today is our demonstration is up to this and in the next lecture we will discuss about the applet programming. And, followed by this discussion, we shall have a demonstration on applet programming.

Thank you.