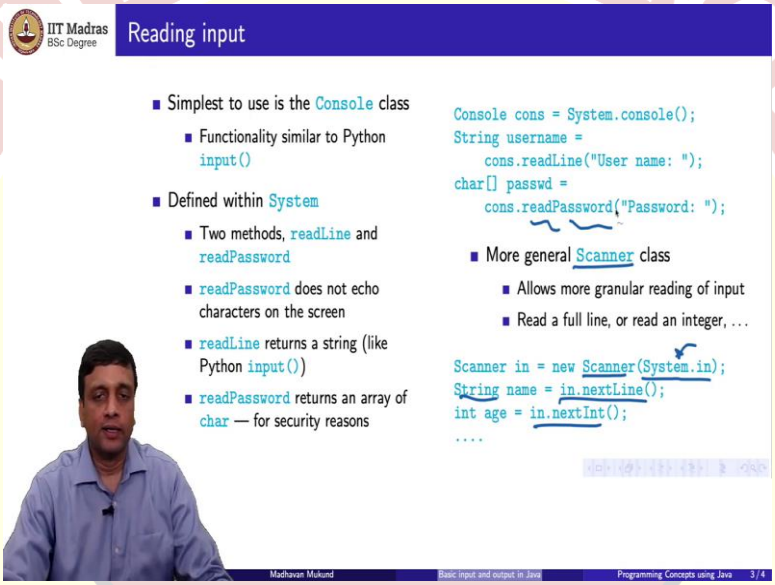

# IIT Madras

## ONLINE DEGREE

**Programming Concepts Using** Java
**Online Degree Programme**
**B. Sc in Programming and Data Science**
**Diploma Level**
**Prof. Madhavan Mukund**
**Department of Computer Science**
**Chennai Mathematical Institute**

**Lecture 11**
**Basic Input and Output in Java**

**(Refer Slide Time: 00:18)**




So, let us conclude this week by looking at some basic input and output in java. So, at the beginning our very first program in java was to print hello world. So, we have an idea about how to print in java. So, we use this println function which takes an argument and prints it out on a new line and we said that system is a public class which has this input output things and in particular it has an output stream.

So, we are printing with respect to that. So, how would we do the converse how would we feed data to a program the simplest way to do this is to use the console class. So, the console class behaves pretty much like the python input function which reads an input from the from the keyboard. So, this is how you do it use you create an instance of the console class and then it has two built-in functions it has a read line and a read password.

So, just like the input function in python the readLine and the read password allow you to display some prompt to the user to indicate what is expected. The difference between

these two is only that when the user types something in response to readLine the user will see it whereas with a password as you know when you type a password or something sensitive in any application it just prints stars or dots or something invisible.

So, a readPassword will not echo the input in a readable form readLine will read there. There is another subtle difference between these two which is that the readLine will return a string as you expect it will return everything until the user presses return. So, the entire line will be read as one string whereas in terms of a password it will be returned as an array of characters.

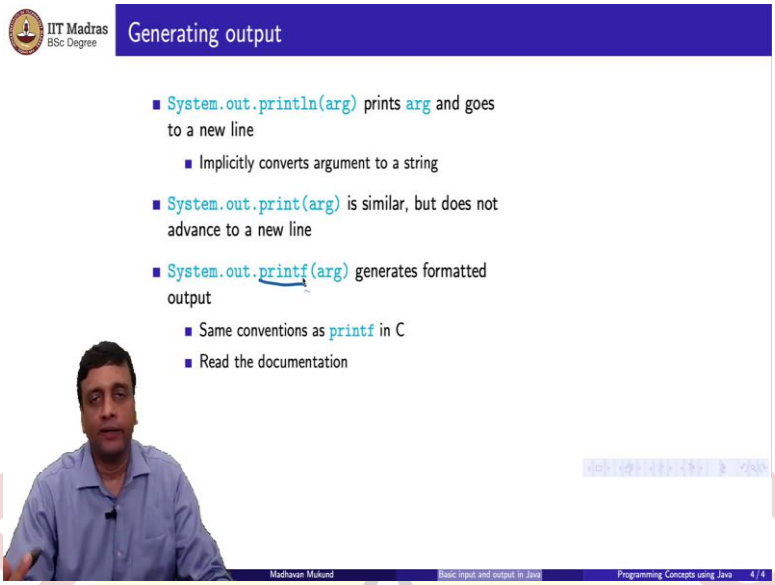
So, it will contain all the individual characters in a character array. So, this is the difference between readPassword and readLine. So, one echoes one does not record one returns a string one returns an array of characters. So, this reads one line at a time and we know from python that then you have to take that string split it convert it into internal that which is a little bit of a nuisance.

So, java does have a more flexible way of doing this using something called a scanner. So, a scanner can be created by. So, remember we had system dot out. So, you can associate with a scanner correspondingly with system dot in. So, you create a new scanner by telling it to scan from the input. So, this is this is you tell the scanner which stream to read from. So, you are telling it to read from system dot in.

And now once you have this scanner you can parse the input in more flexible ways you could of course read the entire line and get a string exactly like the console read line but you can also read the next block as an integer or as different data types. So, you have more granular control over reading the input. So, if you want to be flexible you can use the scanner if you want a very generic way of inputting data to your program use the console class this password reading is not available in the scanner.

So, if you want to read passwords or you want to allow the user to type something which is invisible to the user in that sense some sensitive data then you must use the console class you cannot use the scanner class.

**(Refer Slide Time: 03:38)**




On the other side if you want to display something we already saw the println which is print line which prints the argument and goes to a new line. So, just like the print statement in python what this effectively does is it takes its arguments and convert it to string. So, you can println more than one thing it is not just one argument in general but it could be a sequence of arguments.

So, each of them if it is a string it prints it as a string if it is not a string it will convert it to a representation which can be seen as a string. So, a number 7 will become the character 7 and so on. So, this is what and the key thing is this ln part it goes to a new line. So, if you do not want a new line then there is just a print system dot out dot print which is essentially the same behaviour as system dot out dot println except it will stay on the same line.

So, the next print will continue where this. So, if you want to write things one after the other on the same line then you can use system dot out dot print. And finally if you really want good control over how your values are displayed there is something called printf. Now, this again is something inherited from C. So, printf f stands for formatted if you remember the hello world of the C example that we did in the very first class that use printf.

So, printf stands for formatted output. So, in particular you can say display this integer in two spaces display this floating point with three places after the decimal and so on. So, we will not get into too much detail because it is not something very relevant to the kind

of programming language concepts we want to illustrate in this course but if you read the documentation you can figure out how to use printf to take variables.

So, in python also we had form we have formatted string output you can create a formatted string and then print it. So, it is pretty much the same principle and many of the conventions used for formatting are all inherited from the C, printf. So, they are pretty similar across all these languages but the very specific way to use it you should look up the documentation. So, that you get it right.