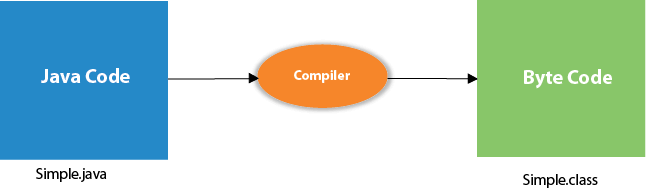
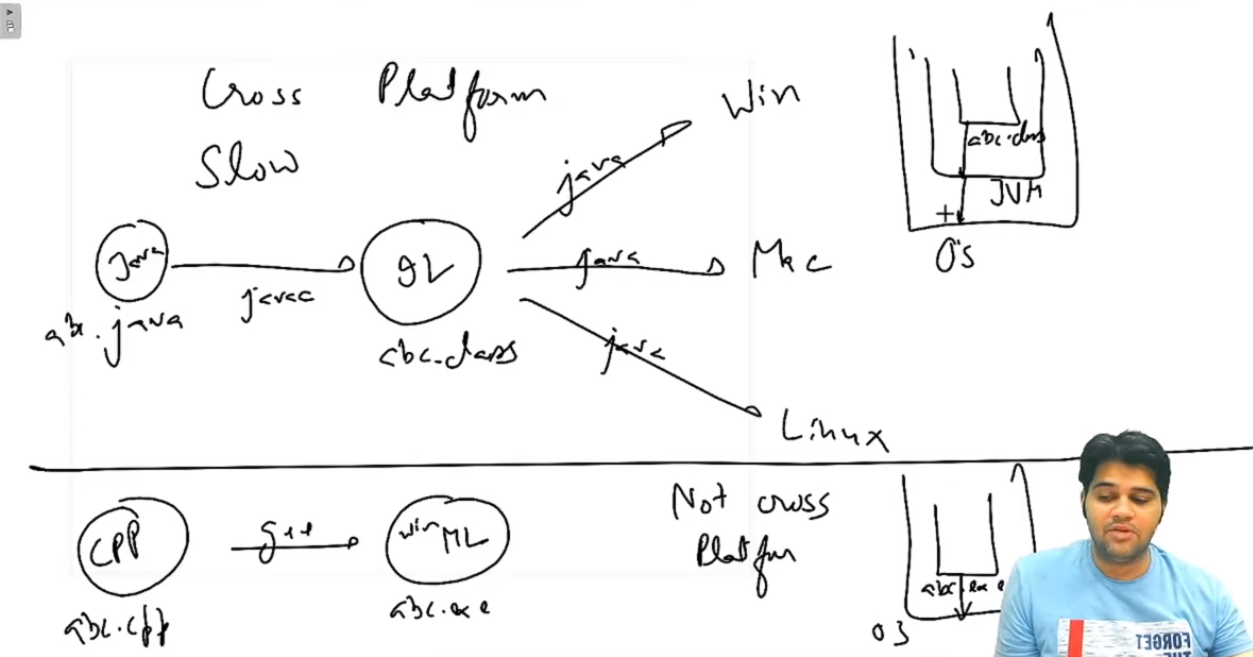
Java for C++ Coders Playlist Link

<https://youtube.com/playlist?list=PL-Jc9J83PIiE0GC3_F-yVem3NJ0bgwfSe>

Notes:

# Lecture 1:





Java Development kit (JDK)

JDK is a software development environment used for making applets and Java applications. The full form of JDK is Java Development Kit. Java developers can use it on Windows, macOS, Solaris, and Linux. JDK helps them to code and run Java programs. It is possible to install more than one JDK version on the same computer.

Why use JDK?

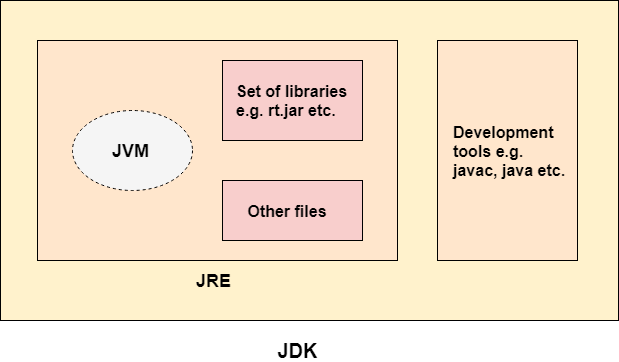
Here are the main reasons for using JDK:

JDK contains tools required to write Java programs and JRE to execute them.

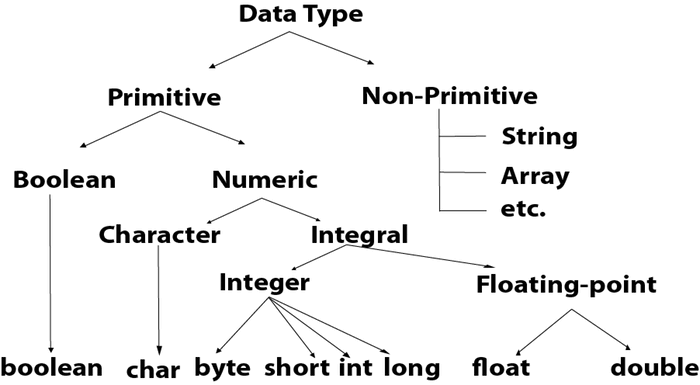
It includes a compiler, Java application launcher, Appletviewer, etc.

Compiler converts code written in Java into byte code.

Java application launcher opens a JRE, loads the necessary class, and executes its main method.



Lecture 3:

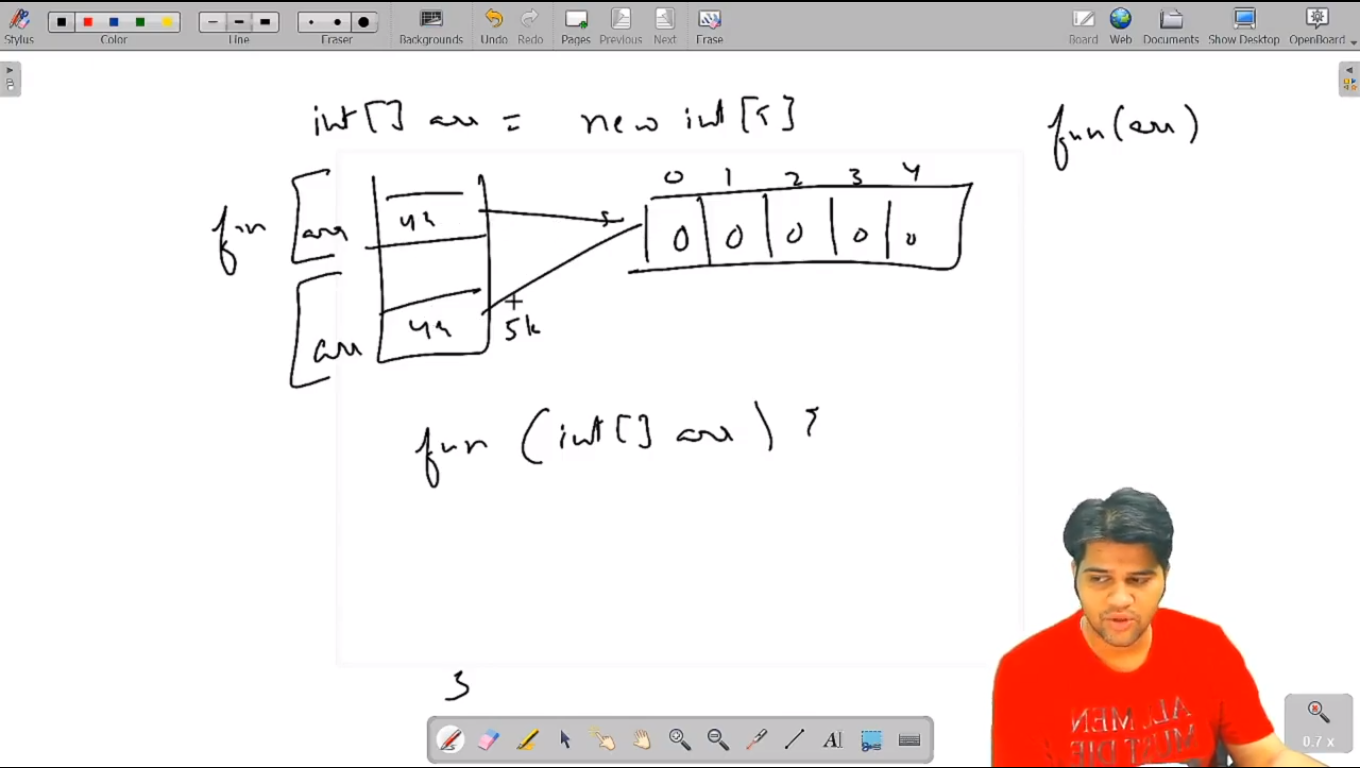


# Arrays

int arr[] --- C style array declaration

int[] arr --- Java style array declaration

Java Always create array in Heap



# 2d Arrays:

In Java, 2d arrays are double pointer -> pointer-> array

i.e. full array in Heap means pointer & array is in heap,

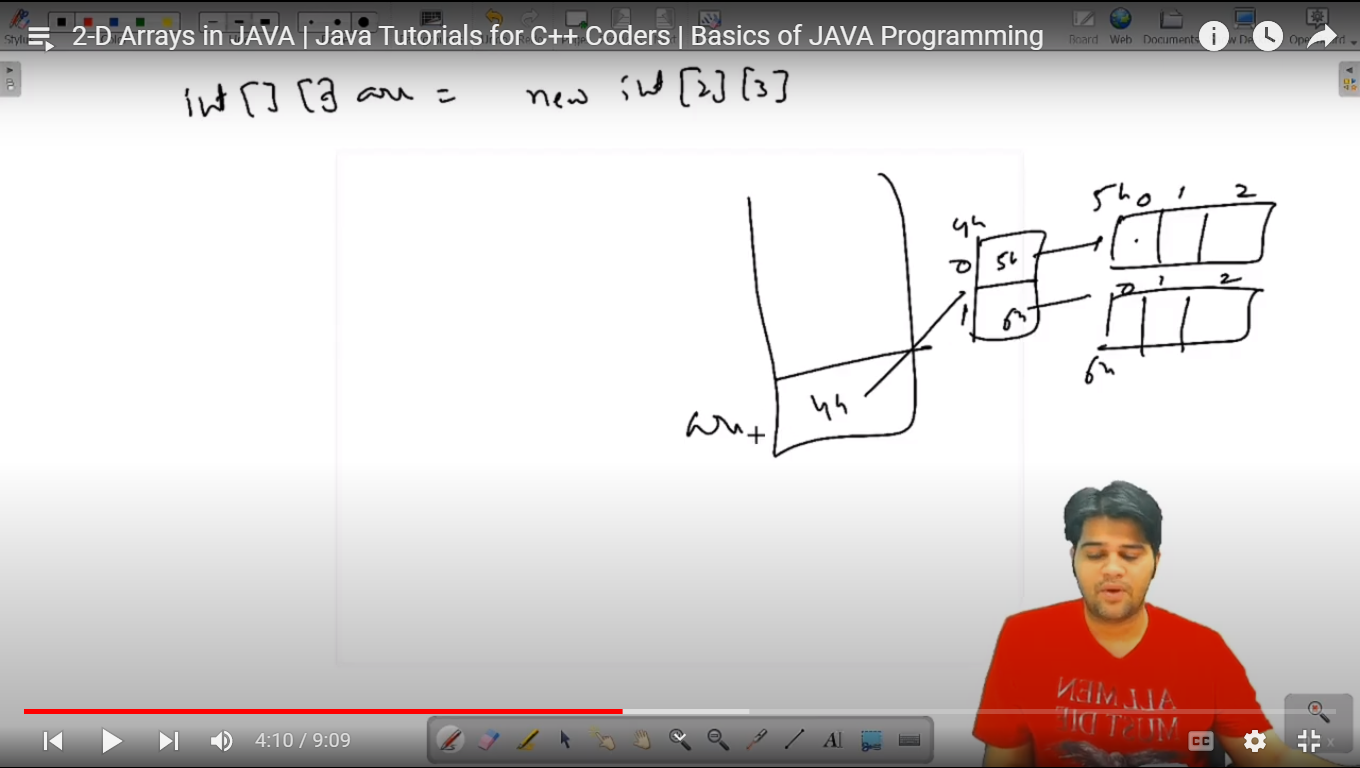
only doble pointer is stored in heap

here double pointer is name of 2d array

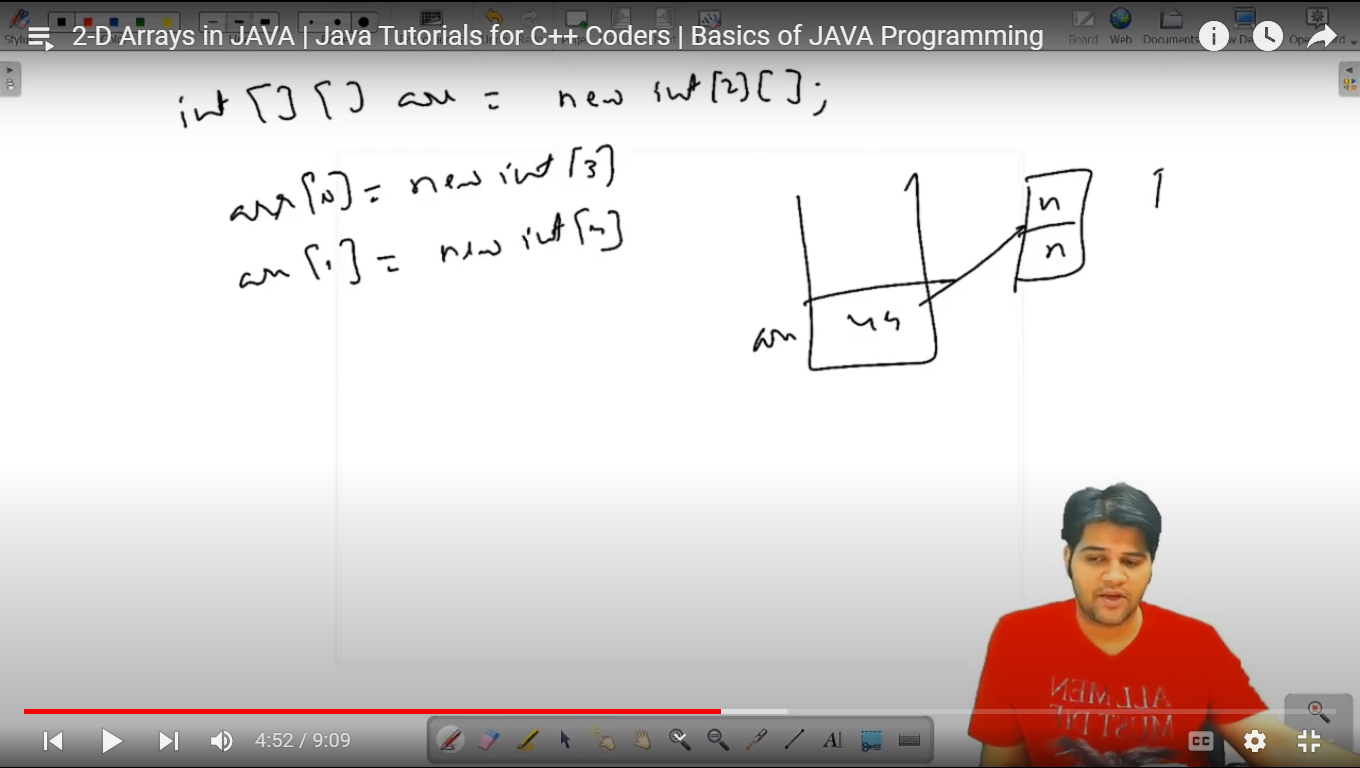
int[][] arr = new int[2][3]; // for fix no of columns

int[][] arr = new int[2][]; // for variable no of columns

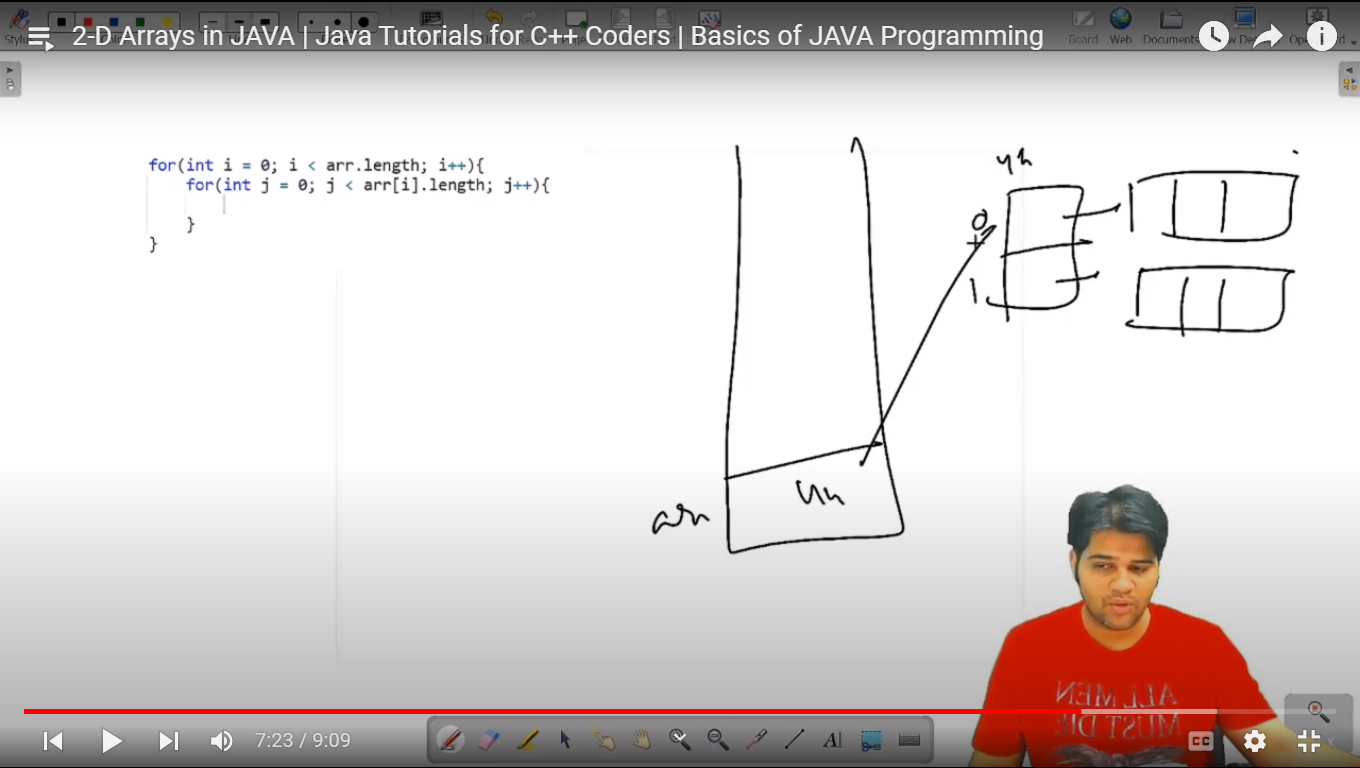
Array having fix no of columns



Jagged Array / Variable Sized array / array having variable no of columns

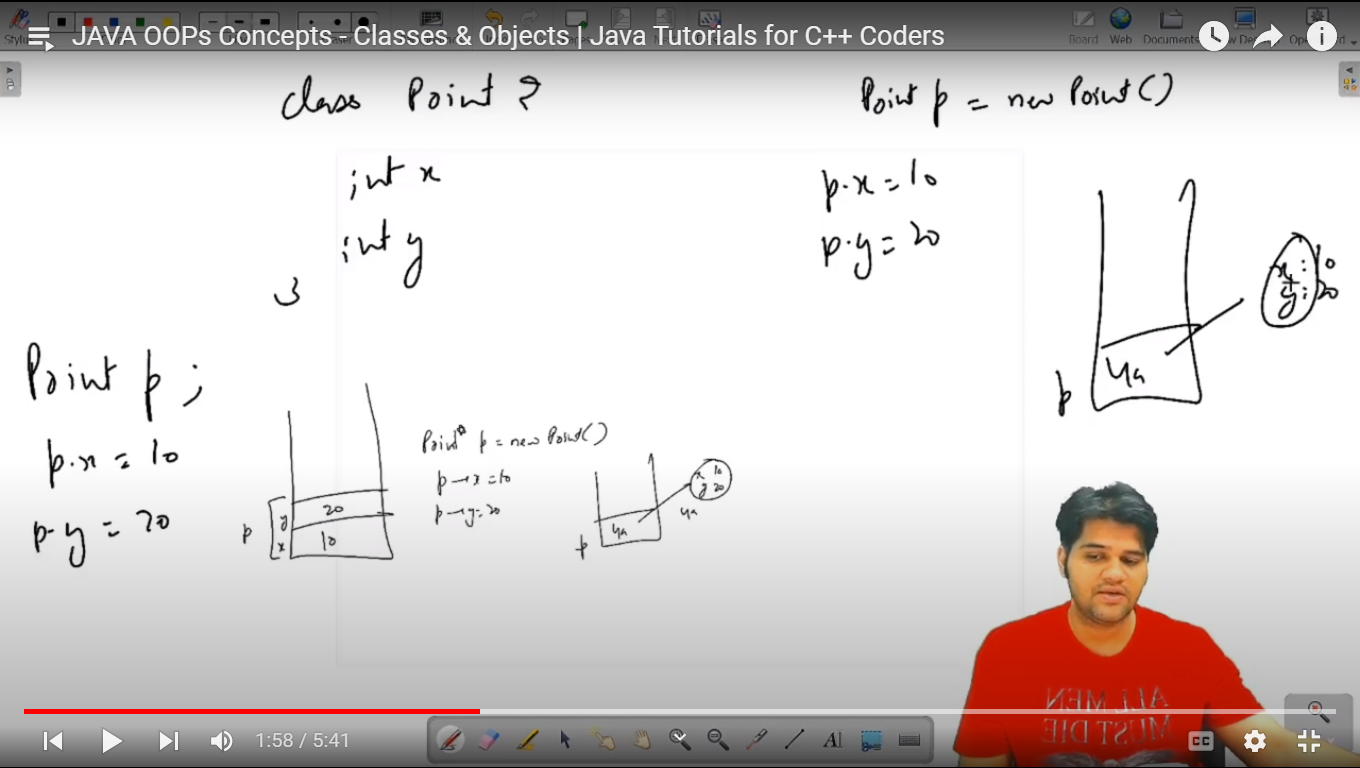


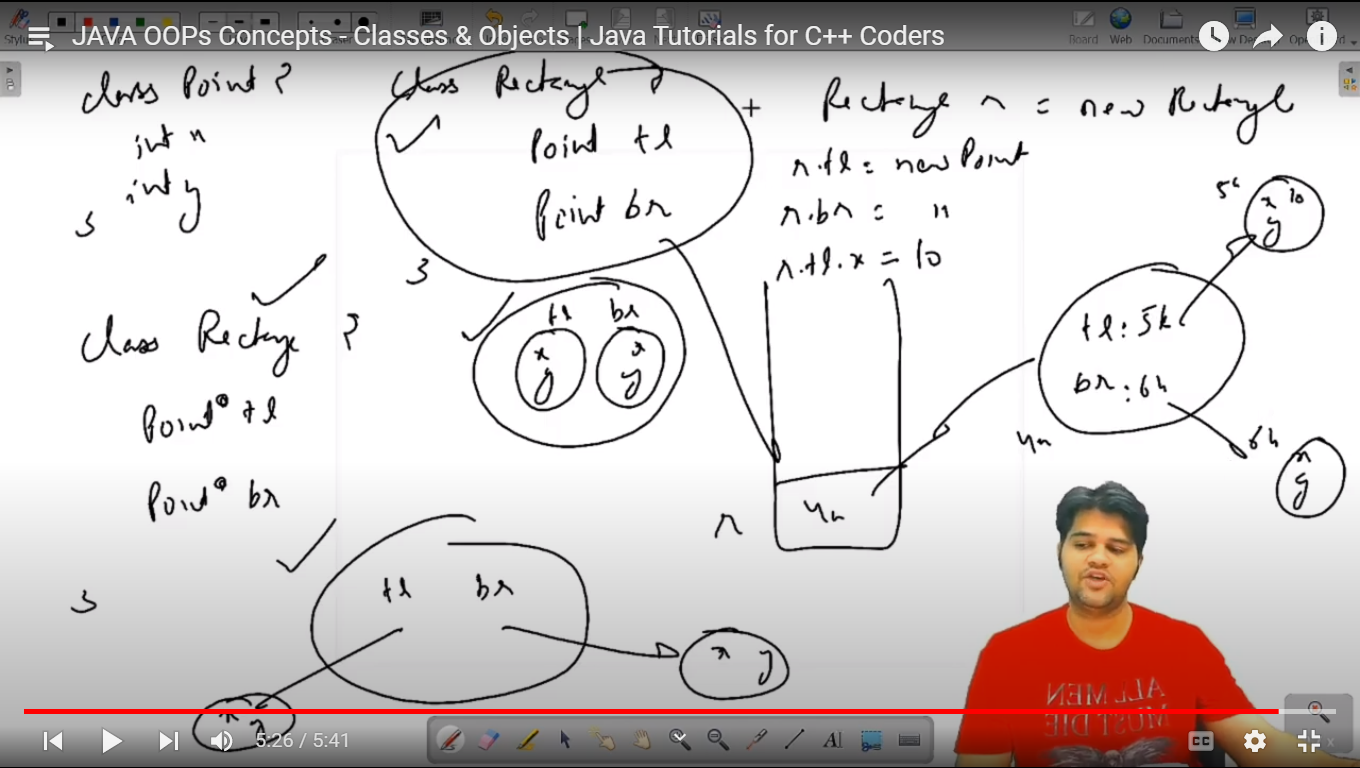
Print 2d array:



# Classes and Objects:

Here we create object in Heap



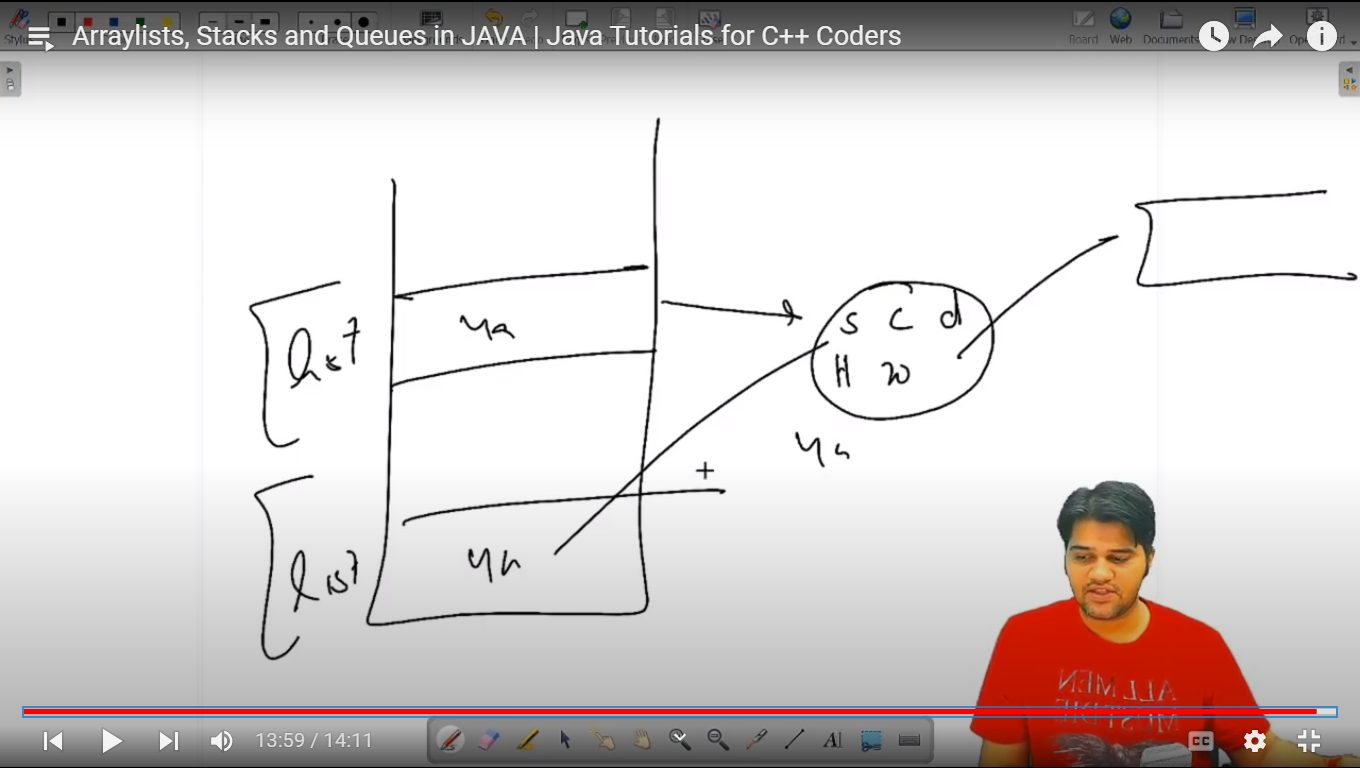


# ArrayList

It is like vector

When we pass arrayList to any function then arraylist will be passed by reference to orginal arraylist

It Doubles its capacity if elements inserted are more than current capacity



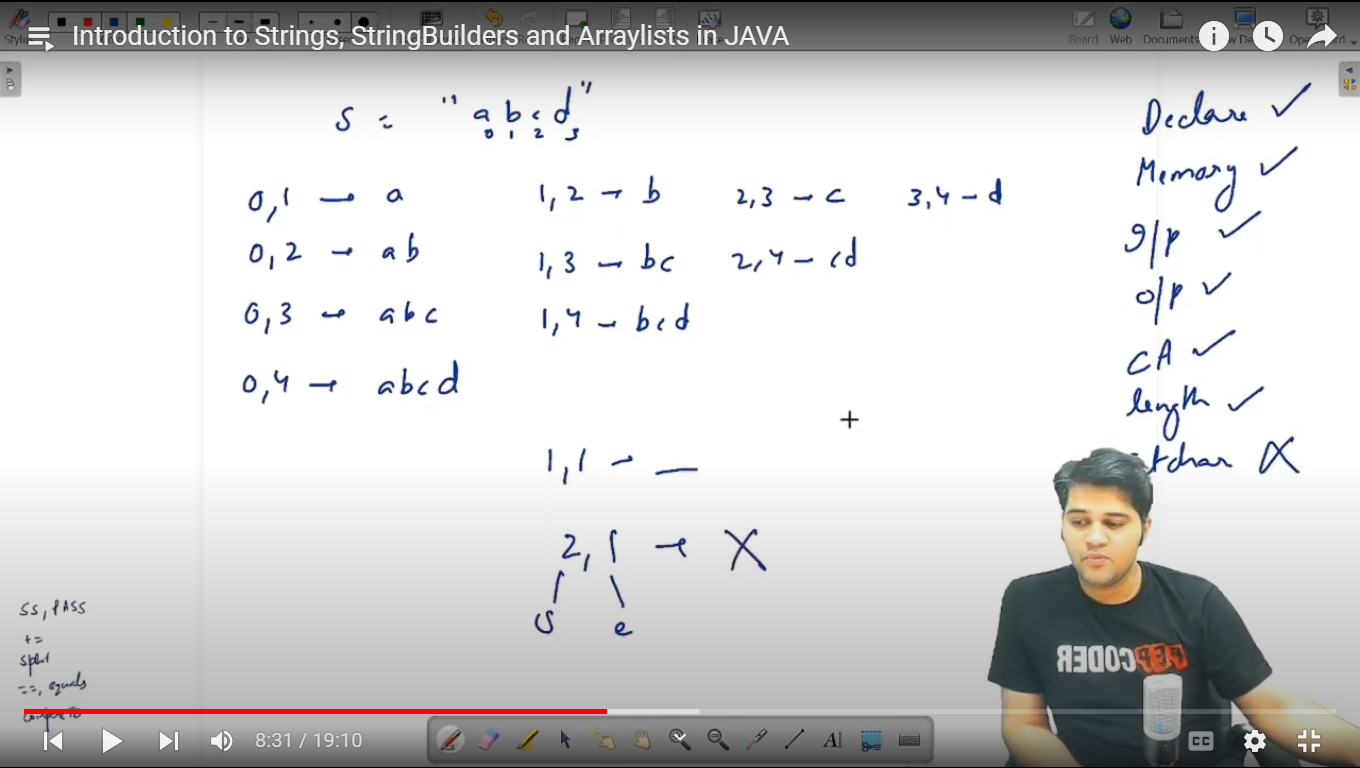
# Strings

In java, Strings are created in Heap by Default

*import* java.util.Scanner;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
  
 *// \*\*\*\*\*\*\* In java, Strings are created in Heap by Default \*\*\*\*\*\*\** Scanner scn = *new* Scanner(System.in);  
  
 *// 1st way to take string input sequentially - Handle & Store enter by using scn.nextLine()* System.out.print("Enter your name = ");  
 String name = scn.next();  
 System.out.println("Your name = "+ name);  
  
 scn.nextLine(); *// It Stores Enter clicked after entering 1st/Above input* System.out.print("Enter your Full Name = ");  
 String fullName = scn.nextLine();  
 System.out.println("Your name = "+ fullName);  
  
 *// 2nd way to take string input sequentially - Use scn.nextLine() only instead of scn.next() for taking string* System.out.print("Enter your name = ");  
 String name1 = scn.nextLine();  
 System.out.println("Your name = "+ name1);  
  
 System.out.print("Enter your Full Name = ");  
 String fullName1 = scn.nextLine();  
 System.out.println("Your name = "+ fullName1);  
  
 *// Link of this issue: https://stackoverflow.com/questions/48475000/why-cant-i-input-two-string-consecutively-in-java  
  
 // Find Length of String --- .length()* System.out.println("Enter job : ");  
 String job = scn.nextLine();  
 System.out.println(job);  
  
 System.out.println(job.length());  
  
 *// get char at particular index --- .charAt(index)  
 char* firstCh = job.charAt(0);  
 System.out.println(firstCh);  
  
 *int* n = job.length();  
 *for*(*int* i=0; i<n; i++){  
 *char* ch = job.charAt(i);  
 System.out.println(ch);  
 }  
  
 *// \*\*\*\* In Java, Strings are immutable, We can't change any character of string \*\*\*\*  
 // \*\*\*\* THERE IS NO setChar() Function in Java \*\*\*\**

job.charAt(1)= ‘R’; // This NOT Works -- WRONG}  
}

# Substring



*// Substring  
 /\*  
 \* .substring(startIdx, endIdx) -> it returns string starting from startIdx to (endIdx - 1)  
 \* Note-> here endIdx is Exclusive & startIdx is Inclusive  
 \*  
 \* .substring(startIdx) -> it returns string starting from startIdx to the end of string  
 \*  
 \* Note-> startIdx <= endIdx always if not then Program will give Error  
 \*  
 \* \*/* String s1 = "abcd";  
  
 System.out.println(s1.substring(0,1));  
  
 *// Below will print Blank space because end index is exclusive* System.out.println(s1.substring(0,0));  
  
 System.out.println(s1.substring(0));  
  
 *// Print All Substrings for give string* System.out.println("Substrings are: ");  
 *for*(*int* i=0; i<s1.length(); i++){  
 *for*(*int* j= i+1; j<=s1.length(); j++){  
 System.out.println(s1.substring(i,j));  
 }  
 }

*// Concatenation* String st1 = "Hi", str2="akash";  
 String message = st1 + " " + str2;  
 System.out.println(message);  
  
 *// you can add char to string  
 // also if you add int to string then internally int will be converted to string while concatenating* String zebra = "White";  
 zebra += ' ';  
 zebra += 'b';  
 zebra += 'l';  
 zebra += 'k';  
 zebra += 10;  
 System.out.println(zebra);  
  
 *// Split() fun* String sentence = "Lorem impsum generator dolor colon";  
 String sentence1 = "Lorem,impsum,generator,dolor colon";  
 String[] words = sentence.split(" ");  
 String[] wordsList1 = sentence1.split(",");  
  
 *for*(*int* i=0 ;i<words.length; i++){  
 System.out.println(words[i]);  
 }  
  
 *for*(String word: wordsList1){  
 System.out.println(word);  
 }

# String Interning and Immutability

* String is immutable means can’t change
* String reference is mutable but instance is not mutable
* Internally when we add a char or new string to old string Then old will points to newly created string in heap
* Interning means we share memory to the same string, It is done to save memory, It is the reason for which setChar() is not allowed in string
* Always use equals() for comparing two strings, equals() compare 1st address of two string is matching or not, If yes then it will return True, Else it 2nd ly check whether content of both string is matching by comparing two strings char by char, if both string content is same Then it will return true Else false
* But == only compare address of two string is matching or not, If yes then it will return True, Else return False
* Strings are slow because every time if new string is added in old string, Then it will create new string and copy old string & then add new string to it

Which results in O(n\*n) if there are n characters are to be added

