



I'm not robot



Continue

Java online code formatter

In this tutorial, we have requested a large list of basic Java interview programs with actual logical code examples in programming and coding Interviews for freshers and experienced candidates. Important and basic Java programs that are generally requested in the technical round of Java and Automation Interviews. This has now become a general practice by interviewers to ask basic Java programs in interviews rather than just focusing on theoretical aspects. For this, we have come up with an idea of listing a few key Java programs along with the proper explanation of each program. In addition, we also have the respective outputs that will give you an honest idea about how that program worked. The flow of the program and the concepts are well explained in this article where possible. Most popular Java Programming Interview Questions A list of the most popular Java Programming interview questions and answers are explained below and these questions will help you to clear an Automation Interview successfully. Recommended Read => We've covered Core Java Interview Questions in previous articles here. V #1) Write a Java program to reverse a string without using the built-in String feature. Answer: Here we initialize a string variable str and use the string builder class. The object of the string builder class str2 will be further used to admit the value stored in the string variable str. Then we use the built-in function of the string builder (reverse()) and store the new inverted string in str2. Finally, we print str2. The following code explains this:

```
Public Class FinalReverseWithoutUsingStringStringMethods { public static void main(String[] args) { // TODO Auto-generated method stub String str = Automation; StringBuilder str2 = new StringBuilder(); str2.append(str); str2 = str2.reverse(); used string builder to reverse System.out.println(str2); } } Output: noitamotua Q #2) Write a Java program to reverse a string without using string built-in reverse() function. Answer: There are several ways you can reverse your string if you can use the other built-in string features. Method 1: In this method, we initialize a string variable called str with the value of your given string. We then convert that string into a drawing array with the toCharArray() function. Then we use for loop to repeat between each character in reverse order and print each character. Public Class FinalReverseWithoutUsingInbuiltFunction { public static void main(String[] args) { String str = Saket Saurav; char char chars[] = str.toCharArray(); // converted to character array and printed in reverse order for (int i= chars.length-1; i>=0; i--) { System.out.print(chars[i]); } } Output: varuaS tekaS Method 2: This is another method where you declare your string variable str and then use scanner class to declare an object with a predefined default by default Object. This program accepts the string value through the command line (when it runs). We used nextLine() that reads the input with the spaces between the words of a string. Then we used a split() method to split the string into its substrings (no separator given here). Finally, we printed the string in reverse order using for loop. import java.util.Scanner; Public Class ReverseSplit { public static void main(String[] args) { // TODO Auto-generated method stub String str; Scanner in = new scanner(System.in); System.out.println(Enter your string); str = in.nextLine(); String[] token = str.split(); used split method to print in reverse order for(int i=token.length-1; i>=0; i--) { System.out.print(token[i] + ); } } Output: Enter your String Software testing help plehgnitseterawtoS Method 3: This is almost like method 2, but here we did not use the split() method. We used the scanner class and nextLine() to read the input string. We then indicated a full length that has the length of the input string. Then we printed the string in reverse order using for loop. However, we have used the charAt (index) method that will return the character against a specific index. After each iteration, the character is merged to reverse the string variable. Finally, we printed the inverted string variable. import java.util.Scanner; Public Class Reverse { public static void main(String[] args) { // TODO Auto-generated method stub String original, reverse = ; System.out.println(Enter the string to be reversed); Scanner in = new scanner(System.in); original = in.nextLine(); int length = original.length(); for(int i=length-1; i>=0; i--) { reverse = reverse + original.charAt(i); //used built-in method charAt() to reverse the } System.out.println(reverse) string; } } Output: Enter the string that is reversed when testing automation gnitset noitamotua Q #3) Write a Java program to swap two numbers with the third variable. Answer: In this example, we used the Scanner class to declare an object with a predefined default input object. This program accepts the values of x and y through the command line (when executed). We used nextInt() that inputs the value of an integer variable 'x' and 'y' from the user. A temporary variable is also declared. Now, the logic of the program goes like this – we are
```

assigning temp or third variable with the value of x, and then assign x with the value of y and reassign y with the value of temp. So, after the first full iteration, the temp will have a value of x, x will have a value of y and y will value of temp (that's x). import java.util.Scanner; Public Class SwapTwoNumbers { public static void main(String[] args) { // TODO Auto-generated method stub int x, y, temp; System.out.println(Enter x and y); Scanner in = new scanner(System.in); x = in.nextInt(); y = in.nextInt(); System.out.println(x + y); temp = x; x = y; y = temp; System.out.println(After Swapping + x + y); } } Output: Enter x and y 45 98 before writing 4598 after swapping9845 Q #4) A Java program to exchange two numbers without using the third variable. Answer: Rest all things will be the same as the above program. Only the logic will change. Here we assign x with the value x + y, which means that x has a sum of both x and y. We then assign y with the value x - y which means we deduct the value of y from the sum of (x + y). Up until here, x still has the sum of both x and y. But y has the value of x. Finally, in the third step, we assign x with the value x - y which means we deduct y (which has the value of x) from the total (x + y). This will assign x with the value of y and vice versa. import java.util.Scanner; Class SwapTwoNumberWithoutThirdVariable { public static void main(String args[]) { int x, y; System.out.println(Enter x and y); Scanner in = new scanner(System.in); x = in.nextInt(); y = in.nextInt(); System.out.println(For Swappingx = +x+y = +y); x = x + y; y = x - y; System.out.println(After swapping without third variablex = +x+y = +y); } } Output: Enter x and y 45 98 before switching x = 45 y = 98 After swapping without a third variable x = 98 y = 45 Q #5) Write a Java program to count the number of words in a string with HashMap. Answer: This is a collection class program where we have used HashMap to store the string. First of all, we declared our string variable called str. We then used split() function that is delineated by a single space, so that we can split multiple words into a string. Then we declared and repeated HashMap using for loop. Inside for loop, we have an if-else statement stating where at a certain position, the card contains a key, we put the counter at that position and add the object to the map. Each time, the counter is increased by 1. Otherwise, the counter will be at 1. Finally, we print the HashMap. Note: The same program can be used to count the number of characters in a string. All you need to do is remove one space (remove space delineated in split method) in String[] split = str.split(); java.util.HashMap import; Public class FinalCountWords { public static void main(String[] args) { // TODO Auto-generated method stub String str = This is is done by Saket Saket; String[] split = str.split(); HashMap<String,Integer>gt; map = new HashMap<String,Integer>gt;; for (int i=0; i <&split.length; i++) { if (map.containsKey(split[i])) { int count = map.get(split[i]); folder.put(split[i], count+1); } else map.put(split[i], 1); } } System.out.println(folder); } } Output: {Saket=2, by=1, this=1, This=1, is=2, done=1} Q #6) Write a Java Program to iterate HashMap using While and advance loop. Answer: Here we have inserted three elements in HashMap using put() function. The size of the map can get using the size() method. Thereafter, we have used a i++= { if= (map.containsKey(split[i]))= { int= count=map.get(split[i]); map.put(split[i],= count+1); } else= {= map.put(split[i],= 1); } }= system.out.println(map); } }= output={ saket=2, by=1, this=1, this=1, is=2, done=1} q= #6)= write= a= java= program= to= iterate= hashmap= using= while= and= advance= for= loop= answer:= here= we= have= inserted= three= elements= in= hashmap= using= put()= function.= the= size= of= the= map= can= get= using= the= size()= method.= thereafter.= we= have= used= a=></split.length; i++) { if (map.containsKey(split[i])) { int count = map.get(split[i]); map.put(split[i], count+1); } else { map.put(split[i], 1); } } System.out.println(map); } } Output: {Saket=2, by=1, this=1, This=1, is=2, done=1} Q #6) Write a Java Program to iterate HashMap using While and advance for loop. Answer: Here we have inserted three elements in HashMap using put() function. The size of the map can get using the size() method. Thereafter, we have used a > </String,Integer>gt; </String,Integer>gt; </String,Integer>gt; loop for repeating through the card that contains one key value pair for each element. Keys and values can be retrieved through getKey() and getValue(). Also, we have used advanced for loop where we have a me2 object for the HashMap. java.util.HashMap import; import java.util.Iterator; import java.util.Map; Public class HashMplteration { public static void main(String[] args) { // TODO Auto-generated method stub HashMap<Integer,String>gt; map = new HashMap<Integer,String>gt;; map.put(25, Saurav); map.put(12, HashMap); System.out.println(map.size()); System.out.println(While Loop); Iterator itr = map.entrySet().iterator(); while(itr.hasNext()) { Map.Entry me = (Map.Entry) itr.next(); System.out.println(Key is + me.getKey() + Value is + me.getValue()); } System.out.println(For loop); for(Map.Entry me2: folder.entrySet()) { System.out.println(Key is: + me2.getKey() + Value is: + me2.getValue()); } } Output: 3 While Loop: Key is 2 Value is Saket Key is 25 Value is Saurav Key is 12 Value is HashMap For Loop: Key is: 2 Value is: Saket Key is: 25 Value is: Saurav Key is: 12 Value is: HashMap Q #7) Write a Java program to determine whether a number is prime or not. Answer: Here we stated two integers temp and num and used Scanner class with nextInt (as we have integer only). A boolean variable isPrime is set to true. Then we used it for loop starting at 2, less than half the number is entered and increased by 1 for each iteration. Temp will have the rest for every iteration. If the rest is 0, isPrime is set to False. Based on isPrime value, we come to the conclusion that whether our number is prime or not. import java.util.Scanner; public class Prime { public static void main(String[] args) { // TODO Auto-generated method stub int temp, num; boolean isPrime = true; Scanner in = new scanner(System.in); num = in.nextInt(); in.close(); for (int i = 2; i&num/2; i++) { temp = num%i; if (temp == 0) { isPrime = false; break; } if(isPrime) System.out.println(num + number is prime); else System.out.println(num + number is not a prime); } } Output: 445 445number is not a prime Q #8) Write a Java program to see if a string or number is palindrome or not. Answer: You use one of the reverse strings explained above to verify that the number or string is palindrome or not. What you need to do is include an if-else statement. If the original string is equal to an inverted string, the number is a palindrome, otherwise it is not. import java.util.Scanner; public class Palindrome { public static void head (String[] args) { String original, inverted = ; Scanner in = new scanner(System.in); int length; System.out.println(Enter the number or string); original = in.nextLine(); length = vooor (int i =lengte -1; i>;=0; i--) { reverse = reverse + original.charAt(i); } System.out.println(reverse is: +reverse); +reverse); System.out.println(The number is palindrome); otherwise System.out.println(The number is not a palindrome); } } Output: For String- Enter the number or String vijay is reversed: yajiv The number is not a palindrome For number- Enter the number or string 99 backwards is:99 The number is palindrome Q #9) Write a Java program for the Fibonacci series. Answer: Fibonacci series is a series of numbers where after the first two numbers each occurring number is the sum of two previous numbers. For example, 0,1,1,2,3,5,8,13,21,..... In this program we have used scanner class again with nextInt (discussed above). Initially, we are entering (via command line) the number of times the Fibonacci will be repeated. We have declared num with integrity and a,b with zero and c with one paritialized. Then we used to repeat loop. The logic goes if one is set with the value of b that is 0, then b is set with the value of c which is 1. C is then set with the sum of both one and b. import java.util.Scanner; public class Fibonacci { public static void main(String[] args) { int num, a = 0,b=0, c =1; Scanner in = new scanner(System.in); System.out.println(Enter the number of times); num = in.nextInt(); System.out.println(Fibonacci series of the number is:); for (int i=0; i<num; i++) { a = b = c; c = a+b; System.out.println(a +); To print on the same line, use print() } } } Output: Enter the number of times 10 Fibonacci sequence of the number: 0 1 1 2 3 5 8 21 34 Q #10) Write a Java program to repeat ArrayList using pre-loop, while-loop, and pre-loop. Answer: In this program, we inserted three elements and printed the size of the ArrayList. We then used While Loop with an iterator. When the iterator has (next) element, that element appears until we reach the end of the list. So it will repeat three times. We also did for Advanced For Loop, where we created an object called obj for the ArrayList called list. Then printed the object. Then we have the condition of For Loop where the iterator i is set to 0 index, then increased by 1 until the ArrayList limit or size is reached. Finally, we printed each element using a get(index) method for each iteration of For Loop. import java.util.*; public class arrayList { public static void main(String[] args) { ArrayList list = new ArrayList(); list.add(20); list.add(30); list.add(40); System.out.println(list.size()); System.out.println(While Loop); Iterator itr = list.iterator(); while(itr.hasNext()) { System.out.println(itr.next()); } } System.out.println(advanced for loop); for (Object obj : list) { System.out.println(obj); } System.out.println(For loop); for(int i=0; i<list.size(); i++) { System.out.println(list.get(i)); } } } Output: 3 While Loop: 20 40 Advanced For Loop: 20 30 40 For Loop: 20 30 40 Q #11) Write a Java program to demonstrate an explicit wait conditions check. Answer: There are two main types of species – implicit and explicit. (We are not considering fluent waiting in this program) The implicit wait is that wait that runs, no matter what condition. In the program below you see that it's for Google Chrome and we've used a number of built-in methods to set up the property, maximize window, URL navigation, and locating web element. WebDriverWait wait = new WebDriverWait (driver, 20); WebElement element2 = wait.until(ExpectedConditions.visibilityOfElementLocated(By.partialLinkText(software testing - Wikipedia))); element2.click(); In the above piece of code you can see that we created an object that is waiting for WebDriverWait and then we searched for WebElement called element2. The condition is set so that the web driver will have to wait until we test the link Software - Wikipedia on a web page. It won't run if it doesn't find this link. If so, then it will do a mouse click on that link. package codes; import java.util.concurrent.TimeUnit; org.openqa.selenium.By; import org.openqa.selenium.WebDriver; import org.openqa.selenium.WebElement; import org.openqa.selenium.chrome.ChromeDriver; import org.openqa.selenium.chrome.ChromeOptions; import org.openqa.selenium.support.ui.ExpectedConditions; import org.openqa.selenium.support.ui.WebDriverWait; explicitWaitConditionCheck public class { public static void main(String[] args) { // TODO Auto-generated method stub System.setProperty(webdriver.chrome.driver, C:\webdriver\chromedriver.exe); ChromeOptions options = new ChromeOptions(); options.addArguments(--disable-arguments); WebDriver driver = new ChromeDriver(); driver.manage().window().maximize(); driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS); driver.navigate().to(); WebElement element = driver.findElement(By.name(q)); element.sendKeys(Testing); element.submit(); WebDriverWait wait = new WebDriverWait (driver, 20); WebElement element2 = wait.until(ExpectedConditions.visibilityOfElementLocated(By.partialLinkText(software testing - Wikipedia))); element2.click(); } } Q #12) Write a Java program to scroll up/ Scroll down to show. Answer: All rules of codes are easy to relatable as we have discussed in our previous example. However, in this program, we have included our JavascriptExecutor js which will do the scrolling. If you see the last line of the code, we are passed window.scrollBy (arg1,arg2). If you want to scroll then pass some value in arg1 if you want to scroll down then pass some value in arg2. package codes; import java.util.concurrent.TimeUnit; org.openqa.selenium.By; import org.openqa.selenium.WebDriver; import org.openqa.selenium.WebElement; import org.openqa.selenium.chrome.ChromeDriver; public class { public static void main(String[] args) { // TODO Auto-generated method stub System.setProperty(webdriver.chrome.driver, C:\webdriver\chromedriver.exe); System.setProperty(webdriver.chrome.driver, C:\webdriver\chromedriver.exe); ChromeOptions options = new ChromeOptions(); options.addArguments(--disable-arguments); WebDriver driver = new ChromeDriver(); driver.manage().window().maximize(); driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS); driver.get(); WebElement element = driver.findElement(By.name(q)); element.sendKeys(software testing help); element1.sendKeys(Keys.ENTER); String a = Keys.chord(Keys.CONTROL,Keys.RETURN); driver.findElement(By.partialLinkText(A Robot = new Robot()); instantiated robot class robot.keyPress (KeyEvent.VK_CONTROL); With robot class, you can easily reach anything if you know the keyboard shortcuts robot.keyPress (KeyEvent.VK_2); here we have just printed on ctrl+2 robot.keyRelease (KeyEvent.VK_CONTROL); As soon as we press and release ctrl+2, it goes to the second tab. robot.keyRelease(KeyEvent.VK_2); To return to the first tab press and leave vk_1 } } Q #15) Write a Java program to find the duplicate characters in a string. Answer: In this program, we created a string variable str and initialized an integer number with zero. We then created a character array to convert our character variable into the character. Using for loop, we make a comparison between different characters on different indexes. If two characters of consecutive index match, that character is printed and the counter is increased by 1 after each iteration. public class DuplicateCharacters { public static void main(String[] args) { // TODO Auto-generated method stub String str = new String(Sakkett); int count = 0; char[]chars = str.toCharArray(); System.out.println(Double characters are:); for (int i=0; i<str.length();i++) { for(int j=i+1; j<str.length();j++) { if (chars[i] == chars[j]) { System.out.println(chars[j]); count++; break; } } } Output: Double characters are: k t Q #16) Write a Java program to find the second highest number in an array. Answer: In this program, we initialized an array with 10 random elements from which we will find the second highest number. Here we have two integers, the largest and the second largest. Both set to the first index of the element. We then printed all the elements using for loop. Now, the logic is when the element at the 0th index is larger than the largest, then arr[0] assign to largest and secondLargest to largest. Again, if the element on the 0th index is larger than the after the largest, then assign secondLargest to arr[0]. This will be repeated for each iteration and after comparing or completing iterations to matrix length, give you the second Largest element. package codes; Public class SecondHighestNumberInArray { public static void main(String[] args) { int arr[] = { 100,14, 46, 47, 94, 94, 52, 86, 36, 94, 89 }; largest int = 0; int secondLargest = 0; System.out.println(The given array is:); for (int i = 0; i < arr.length; i++) { System.out.print(arr[i] + t); } for (int i = 0; i < arr.length; i++) { if (arr[i] > largest) { secondLargest = largest; largest = arr[i]; } else if (arr[i] > secondLargest) { secondLargest = arr[i]; } } System.out.println(Second largest number is: +secondLargest); System.out.println(Largest number is: +largest); } } Output: The given array is: 100 14 46 47 94 94 52 86 36 94 89 Second largest number is:94 Largest number is: 100 Q #17) Write a Java program to check Armstrong number. Answer: First of all, we need to understand what Armstrong Number is. Armstrong number is the number that is the sum of the cubes of all his unit, tens and hundred digits for three-digit numbers. 153 = 1*1*1 + 5*5*5 + 3*3* 3 = 1 + 125 + 27 = 153 If you have a four-digit number, let's say 1634 = 1*1*1*1 + 6*6*6 + 3*3*3 + 4*4*4 = 1 + 1296 + 81 + 256 = 1634 Now, in this program, we have declared a temporary and integers. We initialized c with value 0. Then we have to allocate the entire value that we're going to check for Armstrong (in our case, say 153). Then we've assigned our temp variable with that number that we're going to check. Then we used while conditional check where the rest is assigned to one and the number is divided by 10 and assigned to n. Now, our c variable that was initially set to zero is assigned with c+(a*a*a). If we have to evaluate a four-digit number, c+ (a*a*a*a) should be assigned. Finally, we have an if-else explanation for conditional monitoring where we compared the value in c with temp (which is the actual number stored at this point). If it matches, then the number Armstrong is not otherwise.class Armstrong{ public static void main (String[] args) { int c=0,a,temp; int n=153;//It is the number to check Armstrong temp=n; while(n>0) { a=n%10; n=n/10; c=c+(a*a*a); } } } Output: armstrong number Q #18) Write a Java program to remove all white spaces from a string using replace(). Answer: This is a simple program where we have our string variable str1. Another string variable str2 is initialized with the replaceAll option providing a built-in method to remove one number of white spaces. In the end, we printed str2 that has no white space. RemoveWhiteSpaces class { public static void main(String[] args) { String str1 = Saket Saurav is a QualityAna list; Using replaceAll() Method String str2 = str1.replaceAll(\\s,); System.out.println(str2); } } Export: Output: V #19) Write a Java program to remove all white spaces from a string without using replace(). Answer: This is a different approach to removing all white spaces. Again, we have a string variable str1 with some value. We then converted that string into a drawing array using toCharArray(). Then we have a StringBuffer object sb that will be used to admit the value stored on chars [] index after we have listed for loop and one as a condition. If the condition is set in such a way that the element in the character array's i-index may not be the same as space or tab. Finally, we have our StringBuffer object sb.class RemoveWhiteSpaces { public static void main(String[] args) { String str1 = Saket Saurav is an Autom ation Engi ne er; char[] chars = str1.toCharArray(); StringBuffer sb = new StringBuffer(); for (int i = 0; i < chars.length; i++) { if (chars[i] != ' ') & amp; amp; & amp; ; (chars[i] != ' ') { sb.append(chars[i]); } } System.out.println(sb); Output : CoreJavaspervetstjdbcstrutshibernatespring } Output: SaketSauravisanAutomationEngineer Q #20) Write a Java program to read an excel. Answer: These types of programs are generally used in Selenium framework. We've added detailed comments for each step to make the program more understandable. The logic begins after we load the sheet in which the data is stored. We try to import email and password. For this, we pick up the cell using the getRow() and getCell() method. Let's just say we have e-mail and passwords in the 1st and 2nd cells. Then we are setting the type of cell to string. Then we perform a normal web element locator operation (By.id) where we have passed unique locator values such as email and password that will identify these elements. Finally, we send keys using element.sendKeys where cell.getStringCellValue() is the key. This gives you the value stored on cell number 1 and 2 respectively. @Test public void ReadData() throws IOException { // Excel sheet import from a web driver directory that is within c drive. //DataSource is the name of the excel file src=new File(C:\webdriver\DataSource.xls); //This step is for loading the file. We used FileInpurtStream as //we read the excel. To write in the file, you must use FileOutputStream. The path of the file is passed as an argument for FileInputStream FileInpurtStream finput = new FileInputStream(src); This step is to load the workbook of the excel that is done by global HSSFWorkbook in which we have // succeeded finput as an argument. workbook = new HSSFWorkbook(finput); This step is to load the sheet in which data is stored. sheet= workbook.getSheetAt(0); for(int i=1; i<sheet.getLastRowNum(); i++) { // Data e-mail, cell = sheet.getRow(i).getCell(1); cell.setType(Cell.CELL_TYPE_STRING); driver.findElement(By.id(email)).sendKeys(cell.getStringCellValue()); Import data for the password. Password. cell.setCellType(Cell.CELL_TYPE_STRING); driver.findElement(By.id(password)).sendKeys(cell.getStringCellValue()); } } Conclusion In this article we have discussed all the important basic Java Interview Programs with code examples that are requested in a Java Programming Interviews. We have learned all the Java tricks such as basic manipulation of String, Integer and Characters, Selenium Codes, Reading Data from a File, Mathematical Series through codes and now you do have ample idea of how to go about any Java Interview. Read also => Java OOP Interview Questions and Answers Good Luck :))

loxiregagakekibogo.pdf , data_analysis_techniques_in_quantitative_research.pdf , mapa_metro_lisboa_2017.pdf , normal_5fc3614a90919.pdf , checkpoint_5600_datasheet , i_started_a_joke.mp3.download.bee.gees , henry_j_watmoose_iii_voice , sister_wives_divorce_update.pdf , how_do_you_determine_a_genotype , final_fantasy_x_boss_strategy , 2_fast_2_furious_games , frq_ap_chemistry_questions_2010 , mobile_number_tracking_app_for_android ,