

JAVA

Website to learn basics :

1. <https://www.geeksforgeeks.org/java/>
2. <https://www.tutorialspoint.com/java/index.htm>
3. <https://beginnersbook.com/java-tutorial-for-beginners-with-examples/>
4. <https://www.guru99.com/java-tutorial.html>

Youtube Links :

1. [Java Tutorial For Beginners](#)
2. <https://youtube.com/playlist?list=PL9ooVrP1hQOHb4bxoHauWVwNg4FweDItZ>
3. [Java Essential Training](#)

Puzzles :

[Puzzles](#)

Sample interview Questions :

[Zoho Interview | Set 1 \(On-Campus\)](#)
[Zoho Interview Experience | Set 42\(Off-Campus\)](#)

Basic Programs :

1. Write a Java program to sort a numeric array and a string array.
2. Write a Java program to sum values of an array.
3. Write a Java program to print the following grid.

Expected Output :

4. Write a Java program to calculate the average value of array elements.

5. Write a Java program to test if an array contains a specific value.

6. Write a Java program to find the index of an array element.

7. Write a Java program to remove a specific element from an array.

8. Write a Java program to copy an array by iterating the array.

9. Write a Java program to insert an element (specific position) into an array.

10. Write a Java program to find the maximum and minimum value of an array.

11. Write a Java program to reverse an array of integer values.

12. Write a Java program to find the duplicate values of an array of integer values.
13. Write a Java program to find the duplicate values of an array of string values.
14. Write a Java program to find the common elements between two arrays (string values).
15. Write a Java program to find the common elements between two arrays of integers.
16. Write a Java program to remove duplicate elements from an array.
17. Write a Java program to find the second largest element in an array.
18. Write a Java program to find the second smallest element in an array.
19. Write a Java program to add two matrices of the same size.
20. Write a Java program to convert an array to ArrayList.
21. Write a Java program to convert an ArrayList to an array.
22. Write a Java program to find all pairs of elements in an array whose sum is equal to a specified number.
23. Write a Java program to test the equality of two arrays.

24. Write a Java program to find a missing number in an array.
25. Write a Java program to find common elements from three sorted (in non-decreasing order) arrays.
26. Write a Java program to move all 0's to the end of an array. Maintain the relative order of the other (non-zero) array elements.
27. Write a Java program to find the number of even and odd integers in a given array of integers.
28. Write a Java program to get the difference between the largest and smallest values in an array of integers. The length of the array must be 1 and above.
29. Write a Java program to compute the average value of an array of integers except the largest and smallest values.
30. Write a Java program to check if an array of integers without 0 and -1.
31. Write a Java program to check if the sum of all the 10's in the array is exactly 30. Return false if the condition does not satisfy, otherwise true.
32. Write a Java program to check if an array of integers contains two specified elements 65 and 77.
33. Write a Java program to remove the duplicate elements of a given array and return the new length of the array.

Sample array: [20, 20, 30, 40, 50, 50, 50]

After removing the duplicate elements the program should return 4 as the new length of the array.

34. Write a Java program to find the length of the longest consecutive elements sequence from a given unsorted array of integers.

Sample array: [49, 1, 3, 200, 2, 4, 70, 5]

The longest consecutive elements sequence is [1, 2, 3, 4, 5], therefore the program will return its length 5.

35. Write a Java program to find the sum of the two elements of a given array which is equal to a given integer.

Sample array: [1,2,4,5,6]

Target value: 6.

36. Write a Java program to find all the unique triplets such that sum of all the three elements $[x, y, z \text{ (} x \leq y \leq z \text{)}]$ equal to a specified number.

Sample array: [1, -2, 0, 5, -1, -4]

Target value: 2.

37. Write a Java program to create an array of its anti-diagonals from a given square matrix.

Example:

Input :

1 2

3 4

Output:

[
[1],
[2, 3],
[4]
]

38. Write a Java program to get the majority element from an given array of integers containing duplicates.

Majority element: A majority element is an element that appears more than $n/2$ times where n is the size of the array.

39. Write a Java program to print all the LEADERS in the array.

Note: An element is leader if it is greater than all the elements to its right side.

40. Write a Java program to find the two elements from a given array of positive and negative numbers such that their sum is close to zero.

41. Write a Java program to find smallest and second smallest elements of a given array.

42. Write a Java program to segregate all 0s on left side and all 1s on right side of a given array of 0s and 1s.

43. Write a Java program to find all combination of four elements of an given array whose sum is equal to a given value.

44. Write a Java program to count the number of possible triangles from an given unsorted array of positive integers.

45. Write a Java program to cyclically rotate a given array clockwise by one.
46. Write a Java program to check whether there is a pair with a specified sum of a given sorted and rotated array.
47. Write a Java program to find the rotation count in a given rotated sorted array of integers.
48. Write a Java program to arrange the elements of an given array of integers where all negative integers appear before all the positive integers.
49. Write a Java program to arrange the elements of an given array of integers where all positive integers appear before all the negative integers.
50. Write a Java program to sort an array of positive integers of an array, in the sorted array the value of the first element should be maximum, second value should be minimum value, third should be second maximum, fourth second be second minimum and so on.
51. Write a Java program to separate 0s on left side and 1s on right side of an array of 0s and 1s in random order.
52. Write a Java program to separate even and odd numbers of an array of integers. Put all even numbers first, and then odd numbers.
53. Write a Java program to replace every element with the next greatest element (from right side) in a given array of integers.

Medium:

(Note: In these programs consider a string as array of chars)

Given a list of integers, find the highest product you can get from three of the integers.

Suppose we had a list of n integers sorted in ascending order. How quickly could we check if a given integer is in the list?

Write a function that takes a list of characters and reverses the letters in place.

Write a function `reverse_words()` that takes a message as a list of characters and reverses the order of the words in place.

Paranthesis validation

"Sometimes (when I nest them (my parentheticals) too much (like this (and this))) they get confusing."

Write a function that, given a sentence like the one above, along with the position of an opening parenthesis, finds the corresponding closing parenthesis.

Example: if the example string above is input with the number 10 (position of the first parenthesis), the output should be 79 (position of the last parenthesis).

Logical Advanced:

1. There exists a staircase with N steps, and you can climb up either 1 or 2 steps at a time. Given N , write a function that returns the number of unique ways you can climb the staircase. The order of the steps matters.

For example, if N is 4, then there are 5 unique ways:

1, 1, 1, 1

2, 1, 1

1, 2, 1

1, 1, 2

2, 2

What if, instead of being able to climb 1 or 2 steps at a time, you could climb any number from a set of positive integers X ? For example, if $X = \{1, 3, 5\}$, you could climb 1, 3, or 5 steps at a time.

2. Given a string and a number of lines k , print the string in zigzag form. In zigzag, characters are printed out diagonally from top left to bottom right until reaching the k th line, then back up to top right, and so on.

For example, given the sentence "thisisazigzag" and $k = 4$, you should print:

Out format check this url : <https://imgur.com/a/ERaRQBj>

3. Given an array of integers, return a new array such that each element at index i of the new array is the product of all the numbers in the original array except the one at i . Solve it without using division and in $O(n)$ time.

For example, if our input was $[1, 2, 3, 4, 5]$, the expected output would be $[120, 60, 40, 30, 24]$. If our input was $[3, 2, 1]$, the expected output would be $[2, 3, 6]$.

4. Given a set of integers, return whether the set can be partitioned into two subsets whose sums are the same.

For example, given the set $\{15, 5, 20, 10, 35, 25\}$, it would return true, since we can split the set up into $\{15, 5, 10, 15\}$ and $\{20, 35\}$, which both add up to 55.

Given the set $\{15, 5, 20, 10, 35\}$, it would return false, since we can't split the set up into two subsets that add up to the same sum.

5. Given a N by M matrix of numbers, print out the matrix in a clockwise spiral.

For example, given the following matrix:

[1, 2, 3, 4, 5],

[6, 7, 8, 9, 10],

[11, 12, 13, 14, 15],

[16, 17, 18, 19, 20]]

You should print out the following:

1

2

3

4

5

10

15

20

19

18

17

16

11

6

7

8

9

14

13

12

Advanced Logical :

1. Given a mapping of digits to letters (as in a phone number), and a digit string, return all possible letters the number could represent. You can assume each valid number in the mapping is a single digit. For example if {"2": ["a", "b", "c"], 3: ["d", "e", "f"], ...} then "23" should return ["ad", "ae", "af", "bd", "be", "bf", "cd", "ce", "cf"].

2. Given an array of integers, write a function to determine whether the array could become non-decreasing by modifying at most 1 element. For example, given the array [10, 5, 7], you should return true, since we can modify the 10 into a 1 to make the array non-decreasing. Given the array [10, 5, 1], you should return false, since we can't modify any one element to get a non-decreasing array.

3. Given a list of possibly overlapping intervals, return a new list of intervals where all overlapping intervals have been merged. The input list is not necessarily ordered in any way. For example, given [(1, 3), (5, 8), (4, 10), (20, 25)], you should return [(1, 3), (4, 10), (20, 25)].

4. Given a multiset of integers, return whether it can be partitioned into two subsets whose sums are the same. For example, given the multiset {15, 5, 20, 10, 35, 15, 10}, it would return true, since we can split it up into {15, 5, 10, 15, 10} and {20, 35}, which both add up to 55. Given the multiset {15, 5, 20, 10, 35}, it would return false, since we can't split it up into two subsets that add up to the same sum.

5. There is an N by M matrix of zeroes. Given N and M, write a function to count the number of ways of starting at the top-left corner and getting to the bottom-right corner. You can only move right or down.

6. Given a 2D matrix of characters and a target word, write a function that returns whether the word can be found in the matrix by going left-to-right, or up-to-down.

For example, given the following matrix:

```
[[ 'F', 'A', 'C', 'I'],  
 [ 'O', 'B', 'Q', 'P'],  
 [ 'A', 'N', 'O', 'B'],  
 [ 'M', 'A', 'S', 'S']]
```

and the target word 'FOAM', you should return true, since it's the leftmost column. Similarly, given the target word 'MASS', you should return true, since it's the last row.

7. Given a N by M matrix of numbers, print out the matrix in a clockwise spiral.

8. Given a list of integers, return the largest product that can be made by multiplying any three integers. For example, if the list is [-10, -10, 5, 2], we should return 500, since that's -10 ~~-10~~ 5.

9. A number is considered perfect if its digits sum up to exactly 10. Given a positive integer n, return the n-th perfect number. For example, given 1, you should return 19. Given 2, you should return 28.

10. A left rotation operation on an array of size shifts each of the array's elements unit to the left. For example, if 2 left rotations are performed on [1,2,3,4,5] array, then the array would become [3,4,5,1,2].

Phase-2

OOPs Programming

Create a model/class Car and add data by creating Object.(Notes: list out 5 Car details)

Write class Calculator with constructor ,method add(),sub(),div()and multi().Write class CalculatorDemo from their create instance of Calculator class and try to access its methods.

Write method for below return type:

Returning a void from a method.

Returning a String array from a method.

Returning an int array from a method.

Passing two integer in to method and print out method

Write a model and manager class for Student.

1. Print out all Subject.

2. Print out all Student Names.

3. Top 5 Students

4. Passed Students

5. Failed Students

6. List out all subjects with highest mark.

Write Class MyString Write a method

i) concat(String str) return type String- Concatenates the specified string to the end of this string.

ii) charAt(int index) Returns the char value at the specified index.

iii) endsWith(String suffix) return type boolean-Tests if this string ends with the specified suffix.

iv) static String copyValueOf(char[] data) -Returns a String that represents the character sequence in the array specified.

v) int indexOf(String str)- Returns the index within this string of the first occurrence of the specified substring.

vi) boolean startsWith(String prefix)-Tests if this string starts with the specified prefix.

vii) String substring(int beginIndex)- Returns a new string that is a substring of this string.

Create a class for storing the details of the Donator like donator name,campaign,payment type,etc.

Create a another class for storing the campaign details like launch date, campaign name, list of donator for that campaign,etc

Create a separate class for processing the data of Campaign class and Donator class..

In the main class, Create a menu type structure to add campaigns, add donations, options to list the campaign and donators

All the input should be given during the run time..

Date should be validated, and when the user donates for a campaign, donation date should be between the launch date and deadline..

Campaign

Owner

Campaign Name

Status - Not Started , Running Completed

Launch Date

Fundraiser Deadline

Fundraising Goal

Amount Raised

Description

Event

Campaign - Completed Campaigns - Ongoing Campaigns

Donation

Contributor Name *

Campaign *

Payment Method

Description

Email ID *

Amount * USD

Payment Date [dd-MMM-yyyy]

Evaluate Your Understanding with below questions.

Basics:

Classes and Objects

1. What is the difference between an object-oriented programming language and an object-based programming language?
2. What is encapsulation? How can it be achieved in Java?
3. What is a class?
4. What is an object?
5. Different ways to create object in Java
6. What is a reference?
7. Difference between reference and object.
8. Purpose of constructor.
9. Default values of different data types.

Static:

10. What is the static variable?
11. Difference between instance variable and static variable.

12. What is the static method?
13. Difference between instance method and static method.
14. What are the restrictions that are applied to the Java static methods?
15. Can we override the static methods?
16. What is the static block?
17. Can we execute a program without the main() method?
18. What is the difference between static (class) method and instance method?
19. Can we make the abstract methods static in Java?

'this' keyword:

20. What is 'this' keyword in java?
21. What are the main uses of 'this' keyword?
22. Can we assign the reference to 'this' variable?
23. Can this keyword be used to refer static members?
24. What is constructor chaining? Role of 'this' keyword in this?
25. What are the advantages of passing this into a method instead of the current class object itself?

Inheritance:

26. What is the Inheritance?
27. Advantages of using the inheritance.
28. Which class is the superclass for all the classes?
29. Why is multiple inheritance not supported in java?
30. What is aggregation?
31. What is composition?
32. What is the difference between aggregation and composition?
33. What is Java instanceof operator?

'super' keyword:

34. What is super in java?
35. How can constructor chaining be done by using the super keyword?
36. What are the main uses of the super keyword?
37. What are the differences between this and super keyword?
38. Can you use this() and super() both in the same constructor?

Polymorphism:

39. What are the types of polymorphism?
40. What is method overloading?
41. Rules for method overloading? Justify the rules.
42. Can we overload the methods by making them static?
43. What is method overloading with type promotion?
44. Can you achieve Runtime Polymorphism by data members?
45. What is method overriding?

46. Rules for method overriding? Justify the rules.
47. Can we override the overloaded method?
48. Can we override the private methods?
49. Can we change the scope of the overridden method in the subclass?
50. Can we modify the throws clause of the superclass method while overriding it in the subclass?
51. What is the difference between static binding and dynamic binding?
- 52.

'final' keyword:

53. What is the final variable?
54. What is the final method?
55. What is the final class?
56. What is the final blank variable?
57. Can we initialize the final blank variable?
58. Can we declare a constructor as final?

Abstract class & interfaces:

59. What is the abstraction?
60. What is the abstract class?
61. Can there be an abstract method without an abstract class?
62. Can there be an abstract class without an abstract method?
63. Can you use abstract and final both with a method?
64. Is it possible to create an object for the abstract class?
65. Is it possible to create a reference for the abstract class?
66. What is the interface?
67. Can you declare an interface method static? (Before & after Java8)
68. What is a marker interface? Purpose of a marker interface.
69. What is a functional interface?
70. Is it possible to create an object for the interface?
71. Is it possible to create a reference for the interface?
72. What are the differences between abstract class and interface?
73. Can we define private and protected modifiers for the members in interfaces?
74. When can an object reference be cast to an interface reference?

Access specifiers:

75. What are the different types access specifiers supported in Java?
76. We can declare all the instance variables as public if it so why do we need getter and setter methods?

Wrapper class:

77. What are wrapper classes?
78. What are auto-boxing and unboxing? When does it occur?

Garbage Collection:

79. What is Garbage Collection?
80. Effects using gc()
81. How is garbage collection controlled?
82. How can an object be unreferenced?
83. What is the purpose of the finalize() method?

Packages:

84. What is the package?
85. How can we access some class in another class in Java?
86. What is the static import?

Intermediate:

Nested classes:

1. What is a nested class?
2. What are the advantages of Java inner classes?
3. What are the disadvantages of using inner classes?
4. What are the types of inner classes (non-static nested class) used in Java?
5. Is there any difference between nested classes and inner classes?
6. Can we access the non-final local variable, inside the local inner class?
7. How many class files are created on compiling the OuterClass in the following program?
8. What are anonymous inner classes?
9. What is the nested interface?
10. Can a class have an interface?
11. Can an Interface have a class?

Collection questions :

Basics:

1. List the data structures you are aware of.
2. Why do we need data structures?
3. Relationship between collections and data structures?
4. Do we need really java.util.collections? If it so what are benefits?
5. Hierarchy of collection framework
6. When to choose which collection?

1. List
2. Set
3. Queue
4. Map
7. List the legacy classes and their properties in collection framework.

ist:

8. Why we use List interface? What are main classes implementing List interface?
9. Advantages of using ArrayList over array.
10. Different ways of iterating an ArrayList
11. Difference between ArrayList and LinkedList?
12. Difference between ArrayList and Vector
13. How to convert the array of strings into the list ?
14. What are different ways to iterate over a list?

Set:

15. Why we use Set interface? What are main classes implementing Set interface?
16. Differences Between HashSet, LinkedHashSet and TreeSet
17. User-defined classes in TreeSet
18. What is the difference between Comparable and Comparator interface?
19. Can a null element added to a TreeSet or HashSet?

/span>

20. Why Map interface does not extend Collection interface?
21. Why we use Map interface? What are the main classes implementing Map interface?
22. How to design a good key for HashMap?
23. What classes should one prefer to use a key in HashMap in java?
24. What are different Collection views provided by Map interface?
25. When to use HashMap or TreeMap?
26. Difference between HashMap and Hashtable?
27. Difference between HashMap and HashSet?

Iterators:

28. Difference between Iterator and Enumeration?
29. Difference between Iterator and ListIterator?
30. Why there is not method like Iterator.add() to add elements to the collection?

Others:

31. List the methods and their usages in Collections class.
32. While passing a Collection as argument to a function, how can we make sure the function will not be able to modify it?
33. What does it mean by 'thread-safe'?
34. How to make a collection thread safe?

35. What is UnsupportedOperationException?
36. Which collection classes provide random access of its elements?
37. What is Queue and Stack, List their differences?
38. What are Collections and Arrays class?
39. If an Employee class is present and its objects are added in an ArrayList. Now I want the list to be sorted by the employeeID of Employee class. What are the steps?

Stream:

40. What is a stream? How does it differ from a collection?
41. How can you convert a collection into a stream?

Intermediate:

1. How to reverse the list?
2. How HashSet works internally?
3. How HashMap works internally?
4. What is hash-collision in Hashtable and how it is handled in Java?
5. What is the importance of hashCode() and equals() methods?
6. What is the difference between fail-fast and fail-safe iterators?
7. When will ConcurrentModificationException be thrown?
8. How to avoid ConcurrentModificationException while iterating a collection?
9. What is the difference between intermediate and terminal operations?
10. What is the difference between map and flatMap stream operation?
11. What is stream pipelining in Java 8?

Advanced:

1. Explain ConcurrentHashMap? How it works?
2. What are IdentityHashMap and WeakHashMap?
3. What is BlockingQueue?