Top Java Coding & Conceptual Questions & Answers for SDET Interview Preparation

String reverse. Write a method that will take one string as an argument and return the reverse version of this string.

Solution:

```
package coding;

public class ReverseStr {
   public static void main(String[] args) {
      System.out.println(revStr("apple")); // elppa
      System.out.println(revStr("John")); // nhoJ
      System.out.println(revStr("phone")); // enohp
      System.out.println(revStr("1234567")); // 7654321
   }

   public static String revStr(String str) {
      // create variable to store reversed version of str
      StringBuilder res = new StringBuilder();

      // iterate over input string from the back and use charAt() to get single char
      for (int i = str.length() - 1; i >= 0; i--) {
         res.append(str.charAt(i));
      }

      // convert to string and return reversed version
      return res.toString();
   }
}
```

Array reverse. Write a method that will take an array as an argument and reverse it.

Solution:

```
package coding;

import java.util.Arrays;

public class ReverseArray {
   public static void main(String[] args) {
      int[] inputArr = new int[] {1, 2, 3, 4, 5};
      System.out.println(Arrays.toString(inputArr)); // [1, 2, 3, 4, 5]
      revArr(inputArr);
      System.out.println(Arrays.toString(inputArr)); // [5, 4, 3, 2, 1]
   }

   public static void revArr(int[] arr) {
      // we will use two 'pointers'. One pointer will start from the
beginning
      // another one from the back, and we will swap their values

      int start = 0;
      int end = arr.length - 1;

      while (start < end) {
         // swap elements
         int tmp = arr[start];
         arr[start] = arr[end];
         arr[end] = tmp;

         // increase start and decrease end
         start++;
```

```
            end--;
        }
    }
}
```

Reverse words. Write a method that will take a string as an argument. The method will reverse the position of words and return it.

Solution:

```
package coding;

public class ReverseWords {
    public static void main(String[] args) {
        System.out.println(revWords("apple banana kiwi")); // kiwi
banana apple
        System.out.println(revWords("I am John Doe")); // Doe John am
I
        System.out.println(revWords("orange")); // orange
    }

    public static String revWords(String str) {
        StringBuilder res = new StringBuilder();

        // split input string by " " space to get each word as String[]
        String[] words = str.split(" ");

        // loop over the array from back
        for(int i = words.length - 1; i >= 0; i--) {
            // add words to res with space
            res.append(words[i]).append(" ");
        }
```

```
        // trim needed to remove last space in the end
        return res.toString().trim();
    }
}
```

String palindrome. A palindrome is a word, phrase, number, or sequence of words that reads the same backward as forward.

Solution:

```
package coding;

public class StringPalindrome {
    public static void main(String[] args) {
        System.out.println(isPal("anna")); // true
        System.out.println(isPal("civic")); // true
        System.out.println(isPal("apple")); // false
        System.out.println(isPal("level")); // true
    }

    public static boolean isPal(String str) {
        // we will use two 'pointers'. One pointer will start looking from beginning
        // another from the back. If values of pointers are not equal, we can return false

        int start = 0;
        int end = str.length() - 1;

        while (start < end) {
            // if pointers values are not equal return false
            if(str.charAt(start) != str.charAt(end)){
                return false;
```

```
        }
        start++;
        end--;
    }

    // if program reach here, it means all values were equal, so it's palindrome
    return true;
  }
}
```

Max/min number from an array. Write a method that will accept an array of int as an argument and it returns the max/min number from a given array.

Solution 1:

```
package coding;

public class MaxNumber {
  public static void main(String[] args) {
    System.out.println(max(new int[] {4, 781, 8, 99, 103})); // 781
    System.out.println(max(new int[] {1, 2, 3, 4, 5})); // 5
    System.out.println(max(new int[] {3, 4})); // 4
    System.out.println(max(new int[] {100})); // 100
  }

  public static int max(int[] arrNum) {
    // assume first element of array is the biggest number
    int max = arrNum[0];

    // loop over the array and test our above assumption
    for (int num : arrNum) {
```

```
        // if max was not the biggest number, update it
        if (max < num) {
            max = num;
        }
    }

    // after the loop max variable will hold the biggest number
    return max;
  }
}
```

Solution 2:

```
 package coding;

public class MinNumber {
  public static void main(String[] args) {
    System.out.println(min(new int[] {4, 781, 8, 99, 103})); // 4
    System.out.println(min(new int[] {1, 2, 3, 4, 5})); // 1
    System.out.println(min(new int[] {3, 4})); // 3
    System.out.println(min(new int[] {100})); // 100
  }

  public static int min(int[] arrNum) {
    // assume first element of array is the smallest number
    int min = arrNum[0];

    // loop over the array and test assumption
    for (int num : arrNum) {
      // if min was not smallest, update it
      if (min > num) {
        min = num;
      }
```

```
    }

    return min;
  }
}
```

Static keyword in Java. The static keyword is a very popular question in the interviews.

What's a static keyword in Java?
- Static variables and methods belong to the class, not to a specific object. We need to use static members by class name.

Let's see an example. What's the output of this program? Why this output?

```
public class Person {
  public String name;
  public int age;
  public static String address;

  public static void main(String[] args){
    Person john = new Person();
    john.name = "John";
    john.age = 35;
    john.address = "101 Main St";

    System.out.println(john.name);
    System.out.println(john.age);
    System.out.println(john.address);


    Person smith = new Person();
```

```
    System.out.println(smith.name);
    System.out.println(smith.age);
    System.out.println(smith.address);
  }
}

/*
  Output:
  John
  35
  101 Main St
  null
  0
  101 Main St

  - Static variables belong to class. They do not belong to specific
object.
    That's why for second object print "101 Main St" value for address.
    Correct way of accessing static members is by class name

*/
```

Remove duplicates from a string. Write a method that accepts one string argument and returns it without duplicates. We will see two versions of this method.

Solution:

```
package coding;

import java.util.LinkedHashSet;
import java.util.Set;
```

```
public class RemoveDuplicatesFromStr {
   public static void main(String[] args) {
      System.out.println(removeDup("hello")); // helo
      System.out.println(removeDup("apple")); // aple
      System.out.println(removeDup("aaaaaa")); // a
      System.out.println(removeDup("abc")); // abc
   }

   public static String removeDup(String str) {
      StringBuilder strNoDup = new StringBuilder();

      // loop over string and get each char
      for (char ch : str.toCharArray()) {
         // if strNoDup does not contain char then add to it
         if (!strNoDup.toString().contains(String.valueOf(ch))) {
            strNoDup.append(ch);
         }
      }

      return strNoDup.toString();
   }

   // or do it with Set
   public static String removeDupWithSet(String str) {
      StringBuilder strNoDup = new StringBuilder();

      // convert str to char[]
      char[] letters = str.toCharArray();
      Set<Character> set = new LinkedHashSet<>();

      // add each letter to set. It will remove duplicates - Set does not
allow duplicates
      for (char ch : letters) {
```

```
            set.add(ch);
        }

        // put back to String from Set
        for (Character ch : set) {
            strNoDup.append(ch);
        }

        return strNoDup.toString();
    }
}
```