String High Level Questions

Q1: Write a java program to find the duplicate words and their number of occurrences in a string?
Solution:

```
import java.util.HashMap;
import java.util.Set;
public class DuplicateWordsInString
{
	static void duplicateWords(String inputString)
	{
		String[] words = inputString.split(" ");
		HashMap<String, Integer> wordCount = new HashMap<String, 
Integer>();
		for (String word : words)
		{
			if(wordCount.containsKey(word.toLowerCase()))
			{
				wordCount.put(word.toLowerCase(), 
wordCount.get(word.toLowerCase())+1);
			}else {
				wordCount.put(word.toLowerCase(), 1);
			}
		}
		Set<String> wordsInString = wordCount.keySet();
		for (String word : wordsInString)
		{
			if(wordCount.get(word) > 1)
			{

				System.out.println(word+" : 
"+wordCount.get(word));
			}
		}
	}
	public static void main(String[] args)
	{
		duplicateWords("Hello sathya and sathya");
		duplicateWords("Java is java again java");
		duplicateWords("Super Man Bat Man Spider Man");
	}
}
```

Q 2. Why String is immutable or final in Java?
Answer:

- String pool is possible because string is immutable.
- Since String is immutable it's safe to use in multithreading and we don't need any synchronization.
- Strings are used in java class loader and immutability provides security that correct class is getting loaded by Classloader.

Q 3: Write a java program to count the number of words in a string?
Answer:
```
import java.util.Scanner;

public class CountTheWords {
	public static void main(String[] args)
	{
		System.out.println("Enter the string");
		Scanner sc = new Scanner(System.in);
		String s=sc.nextLine();
		String[] words = s.trim().split(" ");
		System.out.println("Number of words in the string
="+words.length);
	}
}
```

Q 4: Sorting the String without using String API?
Answer:
```
public class SortString {
	public static void main(String[] args) {
		String original = "sathya";
		int j=0;
		char temp=0;
		char[] chars = original.toCharArray();
		for (int i = 0; i <chars.length; i++) {
			for ( j = 0; j < chars.length; j++) {
				if(chars[j]>chars[i]){
					temp=chars[i];
					chars[i]=chars[j];
					chars[j]=temp;
				}
			}
		}
		for(int k=0;k<chars.length;k++){
			System.out.println(chars[k]);
		}

	}
}
```

Q 5. Sort the String with using String API?
```
import java.util.Arrays;
public class SortStringApi
{
	public static void main(String[] args) {
		String original = "sathya";
		char[] chars = original.toCharArray();
		Arrays.sort(chars);
		String sorted = new String(chars);
		System.out.println(sorted);
	}
}
```

Q 6: Java Program to Remove non ASCII chars from String
Solution:
```
public class RemoveNonASCIIString{
	public static void main(String [] args){
		String str = "Instance¡of¥java";
		System.out.println(str);
		str = str.replaceAll("[^\\p{ASCII}]", "");
		System.out.println("After removing non ASCII chars:");
		System.out.println(str);
	}
}
```

Q 7: Java Program to Remove multiple spaces in a string and from normal string with proper spaces?
```
import java.util.StringTokenizer;
public class RemoveSpace{
	public static void main(String [] args){
		String str = "Instance    of    java";
		StringTokenizer st = new StringTokenizer(str, " ");
		StringBuffer sb = new StringBuffer();
		while(st.hasMoreElements()){
			sb.append(st.nextElement()).append(" ");
		}
		System.out.println(sb.toString().trim());
	}
}
```

Q 8: Count the number of words in given string(ex: whitespaces).
```
import java.util.StringTokenizer;

public class StringTokenCount {
	public static void main(String a[]){
		String msg = "This is hello from sathya";
		StringTokenizer st = new StringTokenizer(msg," ");
		System.out.println("Count: "+st.countTokens());
	}
}
```

Q 9: how to break a string based on multiple delimiters. Each character in the constructors delimiter field acts as one delimiter?
Solution:
```
import java.util.StringTokenizer;
public class TokensWithDelimiter {
	public static void main(String a[]){
		String msg = "http://10.123.43.67:80/";
		StringTokenizer st = new StringTokenizer(msg,"://.",true);
		String s="";
		while(st.hasMoreTokens()){
			System.out.println(st.nextToken());
		}
```

```
            System.out.println("Replaced String is
"+msg.replaceAll("[://./]"," "));
      }
}
```

Q 10: Which class does not override the equals() and hashCode() methods, inheriting them directly from class Object?
Answer: java.lang.StringBuffer.

Q 11: What is the difference between StringBuffer and String class ?
Answer: A string buffer implements a mutable sequence of characters. A string buffer is like a String, but can be modified. At any point in time it contains some particular sequence of characters, but the length and content of the sequence can be changed through certain method calls. The String class represents character strings. All string literals in Java programs, such as "abc" are constant and implemented as instances of this class; their values cannot be changed after they are created.

Q 12: Difference between StringBuffer and StringBuilder ?
Answer : StringBuffer is synchronized whereas StringBuilder is not.

Q13: Explain the scenarios to choose between String , StringBuilder and StringBuffer ?
Answer: If the Object value will not change in a scenario use String Class because a String object is immutable. If the Object value can change and will only be modified from a single thread, use a StringBuilder because StringBuilder is unsynchronized(means faster). If the Object value may change, and can be modified by multiple threads, use a StringBuffer because StringBuffer is thread safe(synchronized).

Q 14: Which of the two "StringBuilder" or "StringBuffer" is faster and Why ?
Answer: StringBuilder is not synchronized and hence faster than StringBuffer

Q 15: Java program to check if String is anagram or not?
Solution:

```
import java.util.Arrays;

public class AnagramCheck {
      static void isAnagram(String str1, String str2) {
            String s1 = str1.replaceAll("\\s", "");
            String s2 = str2.replaceAll("\\s", "");
            boolean status = true;
            if (s1.length() != s2.length()) {
                  status = false;
            } else {
                  char[] ArrayS1 = s1.toLowerCase().toCharArray();
                  char[] ArrayS2 = s2.toLowerCase().toCharArray();
                  Arrays.sort(ArrayS1);
                  Arrays.sort(ArrayS2);
                  status = Arrays.equals(ArrayS1, ArrayS2);
            }
            if (status) {
                  System.out.println(s1 + " and " + s2 + " are anagrams");
            } else {
                  System.out.println(s1 + " and " + s2 + " are not 
anagrams");
            }
      }
      public static void main(String[] args) {
            isAnagram("Keep", "Peek");
            isAnagram("Mother In Law", "Hitler Woman");
            isAnagram("Sathya", "syaath");
            isAnagram("Tomoto", "tmotoo");
      }
}
```

Q 16.How to Test immutable in string class?

```
public class Testimmutablestring{
      public static void main(String args[]){
            String s="Sachin";
            s.concat(" Tendulkar");
            System.out.println(s);
            System.out.println(s.concat(" Tendulkar"));
      }
}
```

Q 17: Java Program To Reverse The String With Preserving The Position Of Spaces?
Solution:

```
public class ReverseStringWihtoutApi {
      static void reverseString(String instr) {
            char[] input = instr.toCharArray();
            char[] result = new char[input.length];

            int j = result.length-1;
            for (int i = 0; i < input.length; i++){
                        result[j] = input[i];
                        j--;
            }
            System.out.println(instr+" --->"+String.valueOf(result));
      }
      public static void main(String[] args)
      {
            reverseString("I Am Not String");
            reverseString("JAVA JSP ANDROID");
            reverseString("1 22 333 4444 55555");
      }
}
```

Q 18: What is String pool in java?
Answer: String Pool in Java corresponds to an allocation of memory in Java heap memory. It consists of a collection of String objects, which are shared and reused among several String object references for same String content.

Q 19: Why string class is there when char array is already available?
Answer: String is immutable. Char array is not. A string is implemented with a char array underneath but every time you try to modify it (like with concatenation, replace etc.) it gives you a new String object. So, String behaves as a constant Char array but comes with certain
syntactic sugar that also makes them very easier to use. For example, the addition + operator has been *overloaded* as a string concatenation operator as well.

Q 20: String literal vs String Object.?
Answer: Both expression gives you String object, but there is subtle difference between them. When you use **new String("Hello World!!");** , it explicitly creates a new and referentially distinct instance of a String object. It is an individual instance of the java.lang.String class. **String s="Hello World!!";** may reuse an instance from the string constant pool if one is available (String Pool is a **pool of Strings** stored in Java
heap memory).

Q 21: What is the difference between String, Stringbuilder and StringBuffer?

Ans: String is immutable (once created cannot be changed)object . The object created as a String is stored in the Constant String Pool. Every immutable object in Java is thread safe ,that implies String is also thread safe . String cannot be used by two threads simultaneously. String once assigned cannot be changed.

StringBuffer: StringBuffer is mutable means one can change the value of the object . The object created through StringBuffer is stored in the heap. StringBuffer has the same methods as the StringBuilder , but each method in StringBuffer is synchronized that is StringBuffer is thread safe . Due to this it does not allow two threads to simultaneously access the same method . Each method can be accessed by one thread at a time . But being thread safe has disadvantages too as the performance of the StringBuffer hits due to thread safe property . Thus StringBuilder is faster than the StringBuffer when calling the same methods of each class. String Buffer can be converted to the string by using toString()
method.

StringBuilder: StringBuilder is same as the StringBuffer , that is it stores the object in heap and it can also be modified . The main difference between the StringBuffer and StringBuilder is that StringBuilder is also not thread safe. StringBuilder is fast as it is not thread safe .

Q 22: What is the initial capacity of the following string builder?

StringBuilder sb = new StringBuilder("Able was I ere I saw Elba.");
Answer: It's the length of the initial string + 16: 26 + 16 = 42.

Q 23: How String pool works?
Ans: What happens is that whenever a String literal is encountered in your program, then instead of creating a new String object for that literal, JVM first check whether there is any
String literal *with the same value* already exists in String pool. If there is a string literal with the same value already present in the pool, then *instead of creating new String objec*t for that encountered

String literal in your code, String literal from the pool is used. If there does not exists any string literal with the same value in the pool then a new String object is created, its reference is stored into the string pool and returned.

Q 24: Java program to swap first and last characters of words in a sentence.
Answer:
```
public class SwapFirstLastCharacters {
	static String count(String str)
	{
		char[] ch = str.toCharArray();
		for (int i = 0; i < ch.length; i++) {
			int k = i;
			while (i < ch.length && ch[i] != ' ')
				i++;
			char temp = ch[k];
			ch[k] = ch[i - 1];
			ch[i - 1] = temp;
		}
		return new String(ch);
	}
	public static void main(String[] args)
	{
		String str = "RAGHU SIR FOR JAVA";
		System.out.println(count(str));
	}
}
```

Q 25: Given 3 characters a, b, c. Find the number of strings of length n that can be formed from these 3 characters. Given that : we can use 'a' as many times as we want, 'b' maximum once, and 'c' maximum twice. (Amazon)
Solution:
(----)

Q 26: Print all characters present in given string only once in a reverse order ?
Solution:
```
import java.util.HashSet;
import java.util.Set;
public class StringReverse {
	public static void main(String[] args) {
		String input = "sathyaisbestinhyd";
		reverseWithOneCharacterOnce(input);
	}
	private static void reverseWithOneCharacterOnce(String input) {
		Set<Character> printedChar = new
				HashSet<Character>();
		String reversed = "";
		for (int index = input.length() - 1; index >= 0; index--) {
			Character ch = input.charAt(index);
			if (!printedChar.contains(ch)) {
				printedChar.add(ch);
				reversed = reversed + ch;
			}
```

```
            }
            System.out.println(reversed);
        }
}
```

Q 27: Given two expressions in the form of strings. The task is to compare them and check if they are similar. Expressions consist of lowercase alphabets, '+', '-' and '( )'.

Q 28: **Given a string of lowercase ASCII characters, find all distinct continuous palindromic sub-strings of it**.

Q 29: Given a string and text output the smallest window in the string which covers all the characters present in the text. Both the string and text contains small case letters.
If such window doesn`t exist or this task can not be done then print -1.

Q 30: Given two strings, the task is to find if a string ('a') can be obtained by rotating another string ('b') by two places.

Q 31: Check if the given string **S** is a **Panagram** or not. A pangram is a sentence containing every letter in the English Alphabet.

```
public class Panagram  {
      public static final int n = 26;
      public int check(String arr){
            if(arr.length() < n){
                  return -1;
            }
            for(char c = 'A' ; c <= 'Z' ; c++){
            if((arr.indexOf(c) < 0) && (arr.indexOf((char)(c + 32)) <
                        0)){
                        return -1;
                  }
            }
            return 1;
      }
      public static void main(String[] args) {
            Panagram  obj = new Panagram ();
            int d = obj.check("Abcdefghijklmnopqrstuvwxyz");
            if(d == -1)
                  System.out.print("not pangram");
            else
                  System.out.print("pangram");
      }
}
```

Q 32: What is special about string objects as compared to objects of other derived types?
One special thing about string objects is that you can create string objects without using new operator i.e using string literals. This is not possible with other derived types (except wrapper classes). One more special thing about strings is that you can concatenate two string objects using '+'. This is the relaxation java gives to string objects as they will be used most of the time while coding. And also java provides string constant pool to store the string objects.

Q 33:What is string intern?
Ans: String object in the string constant pool is called as *String Intern*. You can create an exact copy of heap memory string object in string constant pool. This process of creating an exact copy of heap memory string object in the string constant pool is called interning. *intern()* method is used for interning.

Q 34: **What do you think about string constant pool? Why they have provided this pool as we can store string objects in the heap memory itself?**

Ans: String constant pool increases the reusability of existing string objects. When you are creating a string object using string literal, JVM first checks string constant pool. If that object is available, it returns reference of that object rather creating a new object. This will also speed up your application as only reference is returned and also saves the memory as no two objects with same content are created.