Uttara InfoSolutions

www.uttarainfo.com

Uttara Lab - Arrays, Strings!

First compile and run TestStringMethods.java given as example. Go through the code to recap the methods on String we discussed in class and verify if you understand how it is working. In case you have any doubt, ask the Lab Instructor. Methods that you need to use are given in the bottom of the document as well.

1) Write a main() program to test methods of string to perform the following (directly create a string in main() like String str = “abcdef”):
a) check its length
b) print all the chars in string one at a time
c) convert string to array of chars and print chars
d) convert string to uppercase and lowercase and print
e) take 2 strings and check if they are equal (create 2 strings directly in main())
f) take 2 strings and check which is bigger or lesser (alphabetical comparison)
g) take 2 strings and find out if one string occurs in other. Print the first occurring index

2) Create 2 strings as literals with same state and check this:

```
String s1 = "abc";
String s2 = "abc";

System.out.println(s1==s2);
```

Then create a string object using String s3 = new String("abc") and check s1 == s3. Why are you getting true and false as result? Do you understand?
Now check s1.equals(s3) and print the result. What are you getting and why?.

3) String s1 = "happynew";
   s1.concat("year");
   System.out.println(s1); // what is printed? why?

Create StringBuilder sb = new StringBuilder(); Test working of sb.append("abc"), sb.reverse(), sb.toString(), sb.replace() methods.

4) WAM to swap first and last chars of a passed string and return it.
 char at length-1 + substring from 1, length-1 + char at 0

```
str.charAt(str.length()-1) + str.substr(1,str.length()-1) +str.charAt(0)
```

5) WAM to test whether a given string is a palindrome.

6) Test swapping of 2 ints using a) temp variable b) addition & subtraction approach c) swapping using xor (^) approach"

7) WAM to reverse a given string and use this method to a) find whether a string is a palindrome b) reverse each word in a sentence (commonly asked programming question)

Do the reversing of the string in 3 ways - a) using concat by reading each char from end b) using StringBuilder class c) swapping chars in the char array using xor.

8) Create a Math class with following methods:
i) Ability to add 2 ints"
ii) Ability to add n ints (take an array of ints as parameter)"
iii) Ability to add n doubles"
iv) Ability to add 2 complex numbers. Create a Complex class with 2 int instance variables a and b. a represents the real part and b the imaginary.

9) Take any number of strings as input from command line. Concat all

of them and print them out.

10) WAM to accept a string as input and return only unique characters in it (non-repeating) in an array. (commonly asked programming question)”

11) Given a long sequence of limited symbols, Compress the string as shown in the example
i/p : xxxxxxxgggggggggmmckkkllxx"
o/p : x7g9m2c1k3l2x2 (symbol followed by number of occurrences)

12)
See this below code only after trying to solve the problem on your first. This is important.

```
Person
	private String name;
	private int age;
	private String[] petNames = new String[20]; // since he can have max 20 names
	int count=0;

	public void sing()
	{
		String song = "";
		for(int i = 0; i < petNames.length; i++)
		{
			int n = (int)(20 * Math.random());
			song = song + petNames[n];

		}
		SOP(song);
	}
	public void addPetName(String n)
	{
		if(count < petNames.length)
			petNames[count++] = n;
		else
			SOP(..);
	}
	public boolean searchPetName(String n)
```

```
	{
		// search in the petNames array whether a name equal to n exists...and if yes, return true, else return false;

		for(String s : petNames)
		{
			if(s.equals(n))
				return true;
		}
		return false;

	}
```

Understand the correct working of this solution properly. Take assistance of Lab Instructors if required. Ask specific questions.

Methods to use:

Methods required:

```
1) int len = str.length()
2) for (int i = 0 ; i < str.length() ; i++)
	char c = str.charAt(i);
3) char[] arr = str.toCharArray();
4) String s1 = str.toUpperCase(); // str.toLowerCase();
5) boolean result = s1.equals(s2)
6) int pos = s1.indexOf(s2)
7) boolean result = s1.contains(s2)
8) int pos = s1.indexOf(<int pos>, s2) // -1 if search fails
9) String[] sa = str.split(" "); // splits the string on space token
10) int[] arr = {10,20,30}; // int literal array
11) String[] arr = new String[]{"rosey","posey”}; // String literal array
12) for(String s : arr)
	SOP(s);
13) for(int i = 0 ; i < arr.length ; i++)
	SOP(arr[i]);
```

Things to remember:

1) String is a class in Java (capital S)

2) length is a variable in an array and method in a string (arr.length , str.length())
3) You have to capture the returned value from a method to use it: String name = per.getName(); just calling per.getName() will not give you the value.
4) method names of String are fixed. You have to use proper naming convention (camel case).