Uttara InfoSolutions

www.uttarainfo.com

Uttara Lab - Arrays, Strings!

1) WAP to create an array with 10 size and of int holding ability. Store values 1-10 in it. Using the length variable, access each element of the array and print it to the console. Create the array in 2 ways - one using the new operator and then storing the values individually, two by creating the array with the values directly in the array. Loop over the array using an index variable.Also loop over it using for-each loop.

Ex:

```
int[] arr1 = new int[10]; // to create an empty array
int[] arr2 = {10,20,30}; // to create a literal array

for(int i = 0 ; i < arr1.length ; i++) // to loop over the array
        System.out.println("value in "+i+"th box = "+arr1[i]);

for(int val : arr2)
        System.out.println(val);
```

2) Create an array of ints with size 10. Insert 10-1 integer numbers into it (using a for loop). Print out its value to the console using the length variable (in another for loop) and for-each loop.

3) Create an array of ints with size 10. Insert 10 random integer values between 0-100 into it (using a for loop). Print out its value to the console using the length variable (in another for loop) and for-each loop.

4) Create a method called public static void test(int[] arr). Create literal array {10,20,30,40} in main(). Invoke test() and pass this literal array as parameter. Print out its value to the console using the length variable (in another for loop) and for-each loop in test().

5)Code an add() method that will add all the numbers given and return the result (use array of ints as param to the add() method). See AddArrayElements.java if you have a doubt.

Now compile and run TestStringMethods.java given as example (when you run, pass 2 command line inputs like this - java TestStringMethods abcd pqrs). 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. You can also go through other given .java files (TestStringMethods.java, TestMath.java, AddArrayElements.java)

6) 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

7) **Basic String programs(all main() based programs):**

a) Create a String object using new operator and using a string literal and print out its value using SOP to the console

b) Create a String and print out its length to the console using SOP

c) Take the input of a string from the command line (arg[0]) and print "You have a good name, <name concatenated>!" if the length of the string is < 12 chars and "You have a very long name, <name

concatenated>!" if the length of the name is >=12 chars. If no input has been given, show an appropriate error message.

d) Take a string input from arg[0]. Print the chars one at a time to the console.

e) Take a string input from arg[0]. Print the even chars only.

f) Take a string input from arg[0]. Print the alternate chars in one line and the remaining in another.

g) Take string inputs from arg[0], arg[1]. Print "equal" if they have the same contents or "not equal" if they are not.

h) Take string inputs from arg[0], arg[1]. Print whichever string is bigger than the other alphabetically

i) Take string inputs from arg[0], arg[1]. Find out if arg[1] is present in arg[0] string and print if the search succeeds.

j) Take string inputs from arg[0], arg[1]. Convert their cases and print them to the console.

8) WAP in main() to test Math.sqrt(), Math.cbrt(), Math.random(), Math.pow() usage. See TestMath.java for sample code.

9) WAP (directly in main()) to create an array of strings with 5 colors as values. Use Math.random() to randomly print out 5 values from the array.

10) WAM to pass 2 arrays of ints to a method. The method should return the max value present across both the array elements.

11) WAM to pass 2 arrays of ints to a method. The method should return the average of the values across the 2 arrays. The avg returned should be exact and not an approximation.

12) WAM to pass 2 arrays of ints to a method. The method should return the second highest of the values across the 2 arrays.

13) WAP to create a 2 dimensional int array with 3 as first dimension size. Create unidimensional arrays with size 2 and store numbers in the 2-D array. Now loop over the arrays and display the value accordingly.

14) Person has a name and age. A person has a number of pet names(20max) which he obtains over a period of time. He can dance; if his age is less than 25 he can do chacha. If his age is greater than 25, he does the waltz. He can sing too and when he is asked to sing, he uses his petnames to form the song (randomly). Write a tester program to test persons.

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.

15) WAM to pass 2 arrays of ints to a method. The method should return the max value present across both the array elements.

16) WAM to pass 2 arrays of ints to a method. The method should return the average of the values across the 2 arrays. The avg returned should be exact and not an approximation.

17) 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)
```

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

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).