Uttara InfoSolutions

www.uttarainfo.com

Uttara Lab - Arrays, STP & OOP!

Go through other given .java files (TestArray.java, AddArrayElements.java) and understand its working first.

First complete Lab3 programming questions if you have finished them.

Basic Programs to understand working of arrays (you should have gone through Arrays session to do this):

1) Go through TestMath class. There is a class called Math in library using which you can get the square root, cube root, log, sine, cos, tan, floor, ceil, round, random numbers. Use Math class and test the working of these methods as shown.

2) 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);
```

Do you understand the working of for-each loop?

3) 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.

4) 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.

5) 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().

6)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.

OO Programs:

7) (If you have done this already, no need to do it again…go to the next one) There are Dogs. Every Dog has a name and a size. Dogs can bark. If the size of the dog is > 10, it "meows". If the size <=10, then as many times, it "bow wow" its name to the monitor. Test Dog design. After testing the same, make the size variable private and then add setSize()/ getSize() method. See how this impacts your tester class. What check should you add in bark() to ensure that even if the class user has not set size and invokes bark, he gets scolded with a message?

8) Create a class Song. A song has a name (String) and lyrics (String). A song can be played. When you play, it prints out its lyrics to monitor. Create the class with 2 instance variables with 2 setter / getter methods (setName(String str), setLyrics(String n), getName(), getLyrics(). Make sure validate for null and empty string). Now create a TestSong class

with main(). Create 2 objects of Song and set different names and lyrics. Invoke play() and verify if the songs are playing correctly. Now invoke setter methods to change the lyrics and invoke play() again. Has the lyrics changed or not?

In TestSong, create another Song object and set only name. Check by calling play() as to what is the lyrics being printed. Why so? Now invoke setLyrics() and pass a lyrics string as param. Invoke play() again on the same object. Do you now understand why to have setter methods?

9) ***Important problem*** A person has a name, Car, Dog and a favourite Song (reuse classes Car, Dog and Song from earlier problems). When you ask a Person to commute and give him a destination (String parameter), then he will start the car, drive the car and stop the car and print that he has reached the destination. When you ask the person to sing, he will sing (print) his favourite song with lyrics. When you ask a person to take a walk, he will take his dog for a walk and the dog will bark. Person has the ability to eat Food. Food has name and price. Food must be given to Person when you invoke eat(). When a person is asked to eat, he will specify that he is eating food with name and say out his name as well. Person also has a generatePrime() behaviour. When you give him a number as input, then he will generate all prime numbers until that number and print to monitor.

Back to important array problems using StepsToProgram:

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) WAM to accept 2 int arrays as parameters. The method should return one array with only unique elements across both the arrays.

14) 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.

15) ***Important problem**** 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.