April 20th Workshop Summary

For Loops, Arrays and Functions

Functions are officially called methods in java. A method can be thought of as a "input-output" mechanism – they are essentially operations you can perform on your data. Functions are written so that they can take any input of a ***valid*** type and perform the desired outcome.

Why are methods to important? Say we have the following list of integer arrays:

Array 1: [1,5,48,45,1,78,25,2,1,48,6,78,5,25]

Array 2: [1,2,3,4]

Array 3: [1,9,25,4,65,9,9,1]

And we want to find the duplicates in each of these arrays. By writing a method to find duplicates we can give each of these arrays as input to the method. This reduces code redundancy and we can call the same method with different inputs to perform the same operations.

Syntax:

```
public static int methodName(int a, char b) {
   // body
}
```

Here,

public static – modifier. For now you should include this at the beginning of all your methods.

int – return type of the function.

methodName – name of the method.

a, b – formal parameter names

int a, char b – list of parameters (note that the type of the parameter precedes the name of that parameter)

The above mentioned together form your function signature or the "essentials" needed in a working function. Here's **valid** input of this function:

Anytime the function is called it must have EXACTLY 2 inputs

The first input must be an int

The second input must be a char

Eg: methodName (6,'d')

Method Calling

There are 2 ways in which a method can be called. It can either return something or return nothing but modify some value.

```
public static int[] returnDuplicates(int[] listOfNums) {
    int[] outputDuplicateList = new int[listOfNums.length];
    int index =0;
    for (int i=0; i < listOfNums.length-1; i++){
        for (int compareTo = i+1; compareTo < listOfNums.length; compareTo++){
            if (listOfNums[compareTo] == listOfNums[i]){
                outputDuplicateList[index] = listOfNums[compareTo];
                index++;
            }
        }
    }
    return outputDuplicateList;
}

public static void printDuplicates(int[] listNums){
    for (int i =0; i<listNums.length; i++){
        System.out.println(listNums[i]);
    }
}
```

In the image above the method **returnDuplicates** has a return type of **int[].** This method ***returns*** the duplicated numbers as in array of integers.

How it works :

Takes an arrays of integers as input --->

decides which integers are repeated --->

adds them to another new array created inside the function (called a local variable) ---->

outputs this new array

Note that the original input list is unchanged. We are simply copying the values of the duplicates and adding them to a new array which is then returned. We do this by using the keyword “return”.

printDuplicates has an output type of void; i.e it does not have a “return” statement. Instead we are just printing the values of each integer in the input list onto the output window. Print statements are not considered return types and should not be confused as the functions output.

The main function then calls these two function like so:

Summary.java

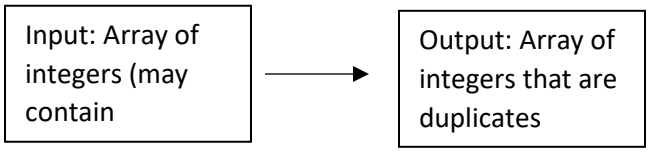
```java
public class Summary {
        public static void main(String args[]) {
            int[] myArray = new int[10];
            myArray[0] = 1;
            myArray[1] = 2;
            myArray[2] = 3;
            myArray[3] = 6;
            myArray[4] = 5;
            myArray[5] = 2;
            myArray[6] = 7;
            myArray[7] = 1;
            myArray[8] = 5;
            myArray[9] = 6;
            int[] duplicates = returnDuplicates(myArray);
            printDuplicates(duplicates);
        }

```

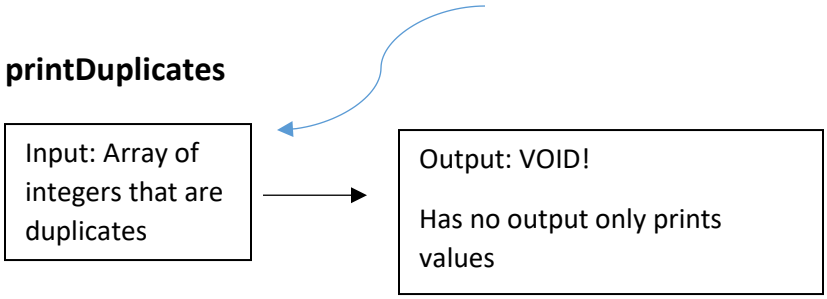
Here the variable duplicates calls the method returnDuplicates to store the array of repeated numbers. This variable is then used as the input for printDuplicates. ***This mechanism is extremely important to understand.***

To summarize :

returnDuplicates



printDuplicates



Being able to write methods that rely on one another is a useful skill for the upcoming weeks and CS in general.