



GIFT School of Engineering and Applied Sciences

Fall 2022

CS-133: Programming Fundamentals

Lab-15 Manual

Methods and Arrays

Task #1: Using Arrays and Methods

In this task, you are being asked to simply declare an array

In a program, create a simple array named hours. Take the array size 5 which is the number of employees. Input the hours worked by employee. Store those hours in array and then simply print those hours.

NOTE: Declare and initialize as well as print the hours in the main method.

1. Create a program called **ArrayLab15.java**.
2. Run **ArrayLab15.java**.
3. Display appropriate messages.

Task #2: Using Arrays and Methods

In this task, you are being asked to write methods that manipulate arrays in Java.

In a program, write a method that accepts two arguments: an array and a number n . Assume that the array contains integers. The method should display all of the numbers in the array that are greater than the number n .

You may use the following header for this method:

```
public static void largerThanNumber(int[] array, int number)
```

1. Create a program called **ArrayMethodsLab15A.java** and add this method in this class.
2. Use a **Scanner** object for the arguments input and for input values in the array.
3. Display appropriate messages.

Task #3: Using Arrays and Methods

In this task, you are being asked to write methods that manipulate arrays in Java.

Write a method that returns the index of the **second occurrence** of an integer value in an integer array. That is, the method searches for the given value for its second occurrence in an array. If that value does not occur more than once, the method returns an index of **-1**.

You may use the following header for this method:

```
public static int getSecondIndex(int[] array, int value)
```

HINTS:

- You will need to count the occurrence of the given value in the array. The method will not stop at finding the first occurrence of the value, and at that point, the count becomes **1**.
- You would continue the search, and when the count becomes equal to **2**, you have found your value.

NOTE: Write methods and the main() method in separate files.

1. Add this method to the **ArrayMethodsLab15B.java** class. Do not create another class.
2. Use a **Scanner** object for the arguments input and for input values in the array.
3. Display appropriate messages based on the return value.

Task #4: Using Arrays and Methods

In this task, you are being asked to write methods that manipulate arrays in Java.

Write a method that returns the index of the ***n*th occurrence** of an integer value in an integer array. That is, the method searches for the given value for its *n*th occurrence, where *n* could be any integer ≥ 1 . If that value does not occur *n* times, the method returns an index of **-1**.

You may use the following header for this method:

```
public static int getNthIndex(int[] array, int occurrence,
int value)
```

For example, the method call:

```
int index = getNthIndex(array, 4, 100);
```

Searches for the **4th occurrence** in the array for the value **100**. If found, the index of the found value is returned, otherwise **-1** is returned.

NOTE:

- **Write methods and the main() method in separate files.**
- You should perform input validation on the **occurrence** argument. It should be greater-than or equal to **1**. If the validation fails, you should print an appropriate error message saying that the **occurrence is invalid!** and return **-1** as the index value.

HINTS:

- You will need to count the occurrence of the given value in the array. The method will not stop at finding the first occurrence of the value, and at that point, the count becomes **1**.
 - You would continue the search, and when the count becomes equal to ***n***, you have found your value.
1. Add this method to the **ArrayMethodsLab15C.java** class. Do not create another class.
 2. Use a **Scanner** object for the arguments input and for input values in the array.
 3. Display appropriate messages based on the return value.

Task #5: Using Arrays and Methods

In this task, you are being asked to write methods that manipulate arrays in Java.

Write a method that returns the index of the **smallest** or the **largest** value in an integer array, based on the given character argument. The character argument could be either 'S' for the smallest, or 'L' for the largest value.

You may use the following header for this method:

```
public static int getSLIndex(int[] array, char typeOfValue)
```

For example, the method call:

```
int index = getSLIndex(array, 'L');
```

Returns the index of the **largest value** in the array.

NOTE:

- **Write methods and the main() method in separate files.**
- You should perform input validation on the typeOfValue argument. It should only be one of 'L' or 'S'. If the validation fails, you should print an appropriate error message saying that the Type of Value is invalid! and return -1 as the index value.

HINTS:

- To find any one of both type of values, you would need to create a variable that holds the smallest value or the largest value and initialize that variable with the **first value** of the array, that is the value at **index 0 of the array**.
 - You would then try to find the smaller or the larger value (as required) from the array, and replace the value stored in the variable with the new-found value.
 - You would continue to do this until you reach the end of the array. At that point, the variable would have the smallest or the largest (as per the argument) value in the array.
1. Add this method to the **ArrayMethodsLab15D.java** class. Do not create another class.
 2. Use a **Scanner** object for the arguments input and for input values in the array.
 3. Display appropriate messages based on the return value.

Task #6: Using Arrays and Methods

In this task, you are being asked to write methods that manipulate arrays in Java.

Write a method that copies an array to another given array in reverse order. Make sure that the lengths of the two arrays are compatible before copying arrays.

You may use the following header for this method:

```
public static void copyReverseArrays(int[] source, int[] target)
```

For example, suppose that the source array has 10 values: **10 9 8 7 6 5 4 3 2 1**

When this method is called with a target array of length 10, the target array would contain values: **1 2 3 4 5 6 7 8 9 10**

NOTE:

- **Write methods and the main() method in separate files.**
1. Add this method to the **ArrayMethodsLab15E.java** class. Do not create another class.
 2. Use a **Scanner** object for the arguments input and for input values in the array.
 3. Display appropriate messages