



Realize Your Career Dreams

GIFT School of Engineering and Applied Sciences

Fall 2018

CS-201: Object-oriented Programming

Lab-1 Manual

Methods and Arrays - Review

Task #1: Writing void Methods

In this task, you are being asked to write void methods in Java.

Write a void method called **getInput()** that takes input using a **Scanner** object, the **name** and **age** from a user. This method calls another method called **printInput()** that takes two arguments, one **String** (for the user name) and another **int** (for the user age) and prints both values using appropriate messages.

You may create method headers as:

```
static void getInput()
```

and

```
static void printInput(String name, int age)
```

NOTE: You need to call the **printInput()** method from the **getInput()** and pass appropriate values. The **main()** method will only call the **getInput()** method. Also, perform input validation on the **age** argument, so that the method should only be called when the **age** is at least **10** and less than **70**.

1. Create a program called **InputMethodLab1.java**.
2. Use a **Scanner** object for the input.
3. Correctly call methods and display appropriate messages.

Task #2: Writing void Methods

In this task, you are being asked to write void methods in Java.

Write a method called **wordsInfo()** that accepts one **String** argument, and prints how many characters are there in that argument, as well as the number of vowels.

You may use the following header for this method:

```
static void wordsInfo(String word)
```

For this exercise, assume that **a e i o u y** are vowels. For example, if the method is called with an argument **"Harry"**, the method prints:

```
wordsInfo("Harry") ;
```

```
//method prints
```

```
5 characters.
```

```
2 vowels.
```

1. Create a program called **CharactersLab1.java**
2. Create appropriate variables and assign values using a **Scanner** object.
3. Correctly display appropriate messages.

Task #3: Writing void Methods

In this task, you are being asked to write void methods in Java.

Write a method called **printTable()** that accepts three positive integer arguments, and then prints the table of the number from the starting number to the ending number.

You may use the following header for this method:

```
static void printTable(int number, int start, int end)
```

NOTE: Perform input validation so that all numbers must be greater than 0 and the “start” number should be less than “end” number.

1. Create a program called **TablesLab1.java**.
2. Create appropriate variables and assign values using a Scanner object.
3. Correctly display appropriate messages.

Task #4: Writing value returning Methods

In this task, you are being asked to write value returning methods in Java.

Write a method called **charAtPosition()** that accepts two arguments: a **String** object, and a positive integer. It then returns the character located at position given by the integer argument.

You may use the following header for this method:

```
static char charAtPosition(String word, int position)
```

For example, if the method is called with the arguments, "GIFT University" and 5, the method returns the character 'U' and a sample output should be:

```
Character at position 5 is: U
```

NOTE: A **String** starts from character position 0. Also, perform input validation so that the **position** must be greater than or equal to 0.

HINT: Use the **charAt(index)** method of the **String** to find the character in a **String**.

1. Create a program called **CharacterPositionLab1.java**.
2. Correctly call the method with appropriate arguments.
3. Correctly display appropriate messages.

Task #5: Writing value returning Method

In this task, you are being asked to write value returning method in Java.

Your job is to write a program which will take two integer numbers as input and returns:

- 0, if both numbers are equal.
- 1, if first number is greater than second number.
- -1, if first number is less than second number.

You may use the following header for this method:

```
static int compare(int firstNumber, int secondNumber)
```

Take input from the user, call the method and print the appropriate messages according to the result.

1. Create a program called **CompareNumberLab1.java**.
2. Create appropriate variables and assign values using a Scanner object.
3. Correctly display appropriate messages.s

Task #6: Writing value returning and void Methods

In this task, you are being asked to write value returning and void methods in Java.

Your job is to write a program which will ask the user for rectangle's length and width, calculates the area of rectangle and display all information of the rectangle with the appropriate messages. Your program should have the following methods:

Value returning methods:

Write a method called **getLength()** that asks the user to enter the length of rectangle and return that value as a double.

Write a method called **getWidth()** that asks the user to enter the width of rectangle and return that value as a double.

Write a method called **getArea()** that calculates the area of rectangle and return that value as a double.

void methods:

Write a method called **display()** that takes the length, width and area of the rectangle as inputs, and then print them with appropriate messages.

NOTE: Perform input validation so that length and width must be greater than 0.

1. Create a program called **RectangleLab1.java**.
2. Create appropriate variables and assign values using a Scanner object.
3. Correctly display appropriate messages.

Task #7: Using Arrays and Methods

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

Write a method called **fillArray()** which will take an integer array as input and the fills the array with the user entered numbers.

You may use the following header for this method:

```
static void fillArray(int[] array)
```

Write another method called **printSumAverage()** which takes an integer array as input and prints the sum and average of the array elements.

You may use the following header for this method:

```
static void printSumAverage(int[] array)
```

Take a number from the user as the size of the array, call the method `fillArray()`, then call the method `printSumAverage()`.

NOTE: Perform input validation so that the size must be greater than 0.

1. Create a program called **ArraySumAverageLab1.java**.
2. Create appropriate variables and assign values using a **Scanner** object.
3. Correctly display appropriate messages.

Task #8: Using Arrays and Methods

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

Write a method called **fillArray()** which will take an integer array as input and the fills the array with the user entered numbers.

You may use the following header for this method:

```
static void fillArray(int[] array)
```

Write a method called **absoluteArray()** which will take an integer array as input and takes the absolute of all the array elements.

You may use the following header for this method:

```
static void absoluteArray(int[] array)
```

Take a number from the user as the size of the array, call the method **fillArray()**, then call the method **absoluteArray()**.

NOTE: Perform input validation so that the array size must be greater than 0.

1. Create a program called **ArrayAbsoluteLab1.java**.
2. Create appropriate variables and assign values using a **Scanner** object.
3. Correctly display appropriate messages.

Task #9: Using Arrays and Methods

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

Write a method that swaps the elements of the array as follows:

- Swap the value of $n-1$ 'th index with the value of 0^{th} index.
- Swap the value of i 'th index with the value of $i+1$ 'th index.

You may use the following header for this method:

```
static int[] arraySwapValues(int[] array)
```

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

The method will return an array which would contain the values: **2 3 4 5 6 7 8 9 10 1**

NOTE: Declare and initialize the array without taking the input from user.

1. Create a program called **ArraySwapValuesLab1.java**.
2. Create appropriate variables and assign values using a **Scanner** object.
3. Correctly display appropriate messages.

Task #10: 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 reverse of a given integer array.

You may use the following header for this method:

```
static int[] reverseArray(int[] array)
```

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

The method will return an array which would contain the values: **1 2 3 4 5 6 7 8 9 10**

NOTE: Declare and initialize the array without taking the input from user.

4. Create a program called **ArrayReverseLab1.java**.
5. Create appropriate variables and assign values using a **Scanner** object.
6. Correctly display appropriate messages.

Task #11: 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 smallest value from a given integer array.

You may use the following header for this method:

```
static int smallest(int[] array)
```

Write another method that returns the largest value from a given integer array.

You may use the following header for this method:

```
static int largest(int[] array)
```

Write a third method that returns the middle value from a given integer array if the array length is odd, or returns the average of the two middle values if the array length is even.

You may use the following header for this method:

```
static double middle(int[] array)
```

NOTE: Declare and initialize the array without taking the input from user.

1. Create a program called **ArrayValuesLab1.java**.
2. Create appropriate variables and assign values using a **Scanner** object.
3. Correctly display appropriate messages.

Task #12: Using Arrays and Methods

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

Write a method that swaps the value of the array in such a way that all negative values comes before the positive values in the array. The method should then returns the array.

You may use the following header for this method:

```
static int[] negativeBeforePositive(int[] array)
```

NOTE: Declare and initialize the array without taking the input from user.

Treat the 0 as a value greater then negative numbers, but lesser than positive numbers.

You may declare some temporary arrays for your help.

1. Create a program called **ArrayNegativeBeforePositiveLab1.java**.
2. Create appropriate variables and assign values using a **Scanner** object.
3. Correctly display appropriate messages.