Lab Guide

*<Java programming language >*

1. *<Session 3 >*
   1. Lab version: 1.0.0
   2. Last updated: 8/9/2023

Overview

In this lab, you will learn all the necessary steps for installing a environment development for Java. You will start with some basic of Java programming, write a simple program to understand some basic features of Java.

**Objectives**

Once you have completed this lab:

* + You will understand how to use for statement
  + You will understand how to use while and do while statement
  + You will able to use array and string in Java
  + You will understand how to sort a array
  + You will understand how to use nested loop in Java

**Exercises**

* 1. This Hands-On Lab is comprised by the following exercises:
  + Oddevensum
  + Compute Pi
  + Do ... While Statement
  + Array
  + Loops And Conditional Statement
  + Nested Loop
  + Break Label
  + Continue Label
  + Array Sort
  + Reverse String
  + Array Of String

Index

[Exercise 1: OddEvenSum 6](#_Toc449620589)

[Task 1 - Create a OddEvenSum Classs 6](#_Toc449620590)

[Task 2 - Write code for OddEvenSum Class 6](#_Toc449620591)

[Task 3 - Execute your program 7](#_Toc449620592)

[Exercise 2: Compute PI 7](#_Toc449620593)

[Task 1 - Create a ComputePI Classs 7](#_Toc449620594)

[Task 2 - Write code for ComputePI Class 7](#_Toc449620595)

[Task 3 - Execute your program 8](#_Toc449620596)

[Exercise 3: Do ... while statement 8](#_Toc449620597)

[Task 1 - Create a DoWhile Classs 8](#_Toc449620598)

[Task 2 - Write code for DoWhile Class 8](#_Toc449620599)

[Task 3 - Execute your program 9](#_Toc449620600)

[Exercise 4: Array 9](#_Toc449620601)

[Task 1 - Create a ArrayTest class 9](#_Toc449620602)

[Task 2 - Write your code inside the main() method 9](#_Toc449620603)

[Task 3 - Execute your program 10](#_Toc449620604)

[Exercise 5: Loops and conditional statement 10](#_Toc449620605)

[Task 1 - Create a PrimeList class 10](#_Toc449620606)

[Task 2 - Write your code inside the main() method 10](#_Toc449620607)

[Task 3 - Execute your program 11](#_Toc449620608)

[Exercise 6: Nested loop 11](#_Toc449620609)

[Task 1 - Create a PrintPattern class 12](#_Toc449620610)

[Task 2 - Write code inside the main() method 12](#_Toc449620611)

[Task 3 - Execute your program 12](#_Toc449620612)

[Exercise 7: Array Sort 13](#_Toc449620613)

[Task 1 - Create a ArrayBubleSort class 13](#_Toc449620614)

[Task 2 - Write code for ArrayBubleSort class 13](#_Toc449620615)

[Task 3 - Execute your program 14](#_Toc449620616)

[Exercise 8: Reverse String 14](#_Toc449620617)

[Task 1 - Create a ReverseString class 14](#_Toc449620618)

[Task 2 - Write code for ReverseString class 14](#_Toc449620619)

[Task 3 - Execute your program 15](#_Toc449620620)

[Exercise 9: Array of String 15](#_Toc449620621)

[Task 1 - Create a Dayofweek class 15](#_Toc449620622)

[Task 2 - Write code for Dayofweek class 15](#_Toc449620623)

[Task 3 - Execute your program 16](#_Toc449620624)

**Next Step**

OddEvenSum

## Create a OddEvenSum Classs

### In the "Package Explorer" (left panel) ⇒ Right-click on your Project (or use the "File" menu) ⇒ New ⇒ Class.

### The "New Java Class" dialog pops up.

* In "Name" field, enter "OddEvenSum ".
* Click "Finish".

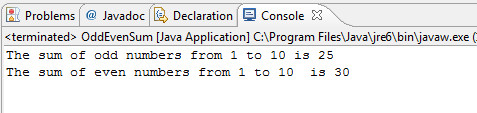
## Write code for OddEvenSum Class

* 1. **public** **class** OddEvenSum {
  2. **public** **static** **void** main(String[] args) {
  3. **int** lowerbound = 1;
  4. **int** upperbound = 10;
  5. **int** sumOdd = 0;
  6. **int** sumEven = 0;
  7. **for** (**int** number = lowerbound; number <= upperbound; number++) {
  8. **if** (number % 2 == 0) {
  9. sumEven += number;
  10. } **else** {
  11. sumOdd += number;
  12. }
  13. }
  14. // Print the result
  15. System.*out*.println("The sum of odd numbers from " + lowerbound + " to "
  16. + upperbound + " is " + sumOdd);
  17. System.*out*.println("The sum of even numbers from " + lowerbound
  18. + " to " + upperbound + "is " + sumEven);
  19. }
  20. }

## Execute your program

### To run the program, right-click anywhere on the source file "OddEvenSum.java" (or from the "Run" menu) ⇒ Choose "Run As" ⇒ "Java Application".

### The output appears on the "Console" panel



1. Compute PI

Write a program called ComputePI to compute the value of π, using the following series expansion. You have to decide on the termination criterion used in the computation (such as the number of terms used or the magnitude of an additional term). Is this series suitable for computing π?

http://www3.ntu.edu.sg/home/ehchua/programming/java/images/ExerciseBasics_ComputePI.png

## Create a ComputePI Classs

### In the "Package Explorer" (left panel) ⇒ Right-click on your Project (or use the "File" menu) ⇒ New ⇒ Class.

### The "New Java Class" dialog pops up.

* In "Name" field, enter "ComputePI ".
* Click "Finish".

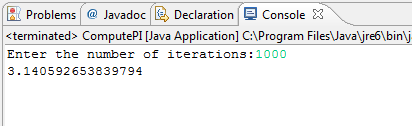
## Write code for ComputePI Class

* 1. **public** **class** ComputePI {
  2. **public** **static** **void** main(String[] args) {
  3. **double** pi = 0;
  4. **double** i = 0;
  5. **int** sign = 1;
  6. Scanner scanner = **new** Scanner(System.*in*);
  7. System.*out*.print("Enter the number of iterations:");
  8. **int** Number\_Of\_Iterations = scanner.nextInt();
  9. **while** (i < Number\_Of\_Iterations) {
  10. pi = pi + 4 \* sign / (2 \* i + 1);
  11. i++;
  12. sign = -sign;
  13. }
  14. System.*out*.println(pi);
  15. }
  16. }

## Execute your program

### To run the program, right-click anywhere on the source file "ComputePI.java" (or from the "Run" menu) ⇒ Choose "Run As" ⇒ "Java Application".

### The output appears on the "Console" panel



Do ... while statement

## Create a DoWhile Classs

### In the "Package Explorer" (left panel) ⇒ Right-click on your Project (or use the "File" menu) ⇒ New ⇒ Class.

### The "New Java Class" dialog pops up.

* In "Name" field, enter "DoWhile ".
* Click "Finish".

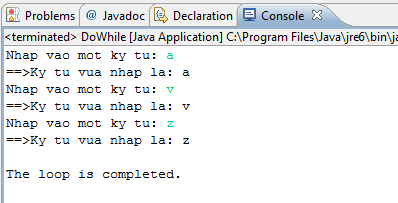
## Write code for Do While Class

* 1. **public** **class** DoWhile {
  2. **public** **static** **void** main(String[] args) {
  3. **char** ch;
  4. Scanner scanner = **new** Scanner(System.*in*);
  5. **do**{
  6. System.*out*.print("Nhap vao mot ky tu: ");
  7. ch = (scanner.next()).charAt(0);
  8. System.*out*.printf("==>Ky tu vua nhap la: %c\n", ch);
  9. }**while**(ch!='z');
  10. System.*out*.print("\nThe loop is completed. \n");
  11. }
  12. }

## Execute your program

### To run the program, right-click anywhere on the source file "DoWhile.java" (or from the "Run" menu) ⇒ Choose "Run As" ⇒ "Java Application".

### The output appears on the "Console" panel



1. Array

## Create a Array Test class

### In the "Package Explorer" (left panel) ⇒ Right-click on your Project (or use the "File" menu) ⇒ New ⇒ Class.

### The "New Java Class" dialog pops up.

* In "Name" field, enter "Arraytest ".
* Click "Finish".

## Write your code inside the main() method

### Declare a array and initiate the values for this array

* 1. **int**[] marks = {74, 43, 58, 60, 90, 64, 70};

### Declare variables to contain sum and average of the array

* 1. **int** sum = 0;
  2. **double** avg;

### Use for loop to calculate sum and average of the array

* 1. **int** count = marks.length;
  2. **for** (**int** i=0; i<count; i++) {
  3. sum += marks[i];
  4. }
  5. avg = (**double**)sum/count;

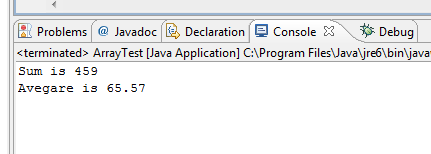
### Print the output to console window

* 1. System.*out*.printf("Sum is %d\n", sum);
  2. System.*out*.printf("Avegare is %.2f\n", avg);

## Execute your program

### To run the program, right-click anywhere on the source file "ArrayTest.java" (or from the "Run" menu) ⇒ Choose "Run As" ⇒ "Java Application".

### The output appears on the "Console" panel

* 1. 

Loops and conditional statement

Write a loop to print all prime number between 2 and 100

## Create a PrimeList class

### In the "Package Explorer" (left panel) ⇒ Right-click on your Project (or use the "File" menu) ⇒ New ⇒ Class.

### The "New Java Class" dialog pops up.

* In "Name" field, enter " PrimeList".
* Click "Finish".

## Write your code inside the main() method

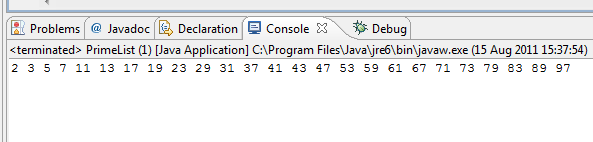
Enter the following codes to find all prime numbers:

* 1. **int** upperbound = 100;
  2. **for** (**int** number = 2; number <= upperbound; number++) {
  3. // Not prime, if there is a factor between 2 and sqrt of number
  4. **int** maxFactor = (**int**)Math.*sqrt*(number);
  5. **boolean** isPrime = **true**;
  6. **int** factor = 2;
  7. **while** (isPrime && factor <= maxFactor) {
  8. **if** (number % factor == 0) { // Factor of number?
  9. isPrime = **false**;
  10. }
  11. factor++;
  12. }
  13. **if** (isPrime)
  14. System.*out*.print(number + " ");
  15. }

## Execute your program

### To run the program, right-click anywhere on the source file "PrimeList.java" (or from the "Run" menu) ⇒ Choose "Run As" ⇒ "Java Application".

### The output appears on the "Console" panel

* 1. 

Nested loop

In this exercise, you will use a **nested loop** to print this pattern:

########

#######

######

#####

####

###

##

#

## Create a PrintPattern class

### In the "Package Explorer" (left panel) ⇒ Right-click on your Project (or use the "File" menu) ⇒ New ⇒ Class.

### The "New Java Class" dialog pops up.

* In "Name" field, enter "PrintPattern".
* Click "Finish".

## Write code inside the main() method

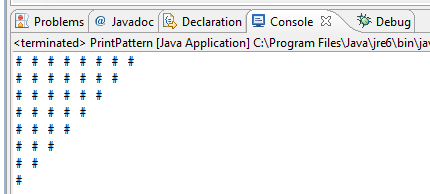
Enter the following codes to print this pattern

* 1. **int** size = 8;
  2. **for** (**int** row = 1; row <= size; row++) {
  3. **for** (**int** col = row; col <= size; col++) {
  4. System.*out*.print("# ");
  5. }
  6. System.*out*.println();
  7. }

## Execute your program

### To run the program, right-click anywhere on the source file “Printpattern.java" (or from the "Run" menu) ⇒ Choose "Run As" ⇒ "Java Application".

### The output appears on the "Console" panel



Array Sort

This program will use bubble sort algorithm

## Create a ArrayBubleSort class

### In the "Package Explorer" (left panel) ⇒ Right-click on your Project (or use the "File" menu) ⇒ New ⇒ Class.

### The "New Java Class" dialog pops up.

* In "Name" field, enter "ArrayBubleSort".
* Click "Finish".

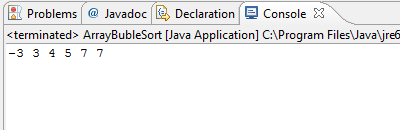
## Write code for ArrayBubleSort class

* 1. **public** **class** ArrayBubleSort {
  2. **public** **static** **void** main(String[] args) {
  3. **int** temp;
  4. **int**[] a = { 4, 5, 3, 7, -3, 9, 7 };
  5. **for** (**int** i = 0; i < a.length - 1; i++) {
  6. **for** (**int** j = i; j <= a.length - 1; j++) {
  7. **if** (a[i] > a[j]) {
  8. temp = a[i];
  9. a[i] = a[j];
  10. a[j] = temp;
  11. }
  12. }
  13. }
  14. **for** (**int** i = 0; i < a.length - 1; i++) {
  15. System.*out*.print(a[i] + " ");
  16. }
  17. }
  18. }

## Execute your program

### To run the program, right-click anywhere on the source file “ArrayBubleSort.java" (or from the "Run" menu) ⇒ Choose "Run As" ⇒ "Java Application".

### The output appears on the "Console" panel



Reverse String

Write a program to reverse a string. It invokes the String method charAt(i), which returns the ith character in the string, counting from 0.

## Create a ReverseString class

### In the "Package Explorer" (left panel) ⇒ Right-click on your Project (or use the "File" menu) ⇒ New ⇒ Class.

### The "New Java Class" dialog pops up.

* In "Name" field, enter "ReverString".
* Click "Finish".

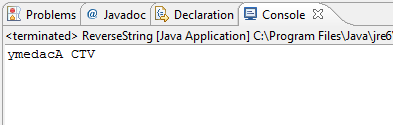
## Write code for ReverseString class

* 1. **public** **class** ReverseString {
  2. **public** **static** **void** main(String[] args) {
  3. String palindrome = "Java Academy";
  4. **int** len = palindrome.length();
  5. **char**[] tempCharArray = **new** **char**[len];
  6. **char**[] charArray = **new** **char**[len];
  8. // put original string in an array of chars
  9. **for** (**int** i = 0; i < len; i++) {
  10. tempCharArray[i] = palindrome.charAt(i);
  11. }
  13. // reverse array of chars
  14. **for** (**int** j = 0; j < len; j++) {
  15. charArray[j] = tempCharArray[len - 1 - j];
  16. }
  18. String reversePalindrome = **new** String(charArray);
  19. System.*out*.println(reversePalindrome);
  20. }
  21. }

## Execute your program

### To run the program, right-click anywhere on the source file “ReverseString.java" (or from the "Run" menu) ⇒ Choose "Run As" ⇒ "Java Application".

### The output appears on the "Console" panel

* 1. 

1. Array of String

Write a program to take a number and return the day of the week

## Create a Dayofweek class

### In the "Package Explorer" (left panel) ⇒ Right-click on your Project (or use the "File" menu) ⇒ New ⇒ Class.

### The "New Java Class" dialog pops up.

* In "Name" field, enter "DayofWeek".
* Click "Finish".

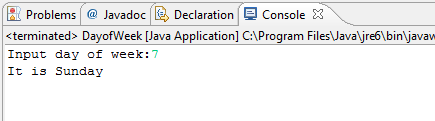
## Write code for Dayofweek class

* 1. **public** **class** DayofWeek {
  2. **public** **static** **void** main(String[] args) {
  3. Scanner scanner = **new** Scanner(System.*in*);
  4. String[] calendarDays = { "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday" };
  5. System.*out*.print("Input day of week:");
  6. **int** dayofweek = scanner.nextInt();
  7. System.*out*.println("It is " + calendarDays[dayofweek - 1]);
  8. }
  9. }

## Execute your program

### To run the program, right-click anywhere on the source file “DayofWeek.java" (or from the "Run" menu) ⇒ Choose "Run As" ⇒ "Java Application".

### The output appears on the "Console" panel



Summary

* 1. In this lab, you have learned how to use loop and jump statement, array, string in Java