



## Lab04

### Problem Solving Skill & Basic Java Arithmetic Operations

#### **Submission Details**

In this lab, you are required to write a Java program (SeparateDigits.java) to solve the given problems shown in the section “Lab Exercise”. After you have completed the implementation using **Eclipse**, submit the program file to the assignment page of Canvas. The deadline would be **ONE week** after your lab session conducted. You are reminded to double-check your solution to verify everything in correct before submission.

Note: You are required to put your name, student ID and lab section at the beginning of your source file as the comment.

#### **Objective**

The objectives of this lab are (1) to familiar with the basic problem solving skill; (2) to reinforce the basic concept of user inputs from keyboard and basic arithmetic operations.

#### **Software Required**

1. Java Development Kit, eg JavaSE 1.8
2. An Integrated Development Environment, e.g. Eclipse

Please start your PC with Window 7 and refer to the previous lab exercise for the installation and operation guide of Eclipse!

#### **Introduction**

Similar to lab 3 In this lab, you will practice:

1. To obtain input from the user using the Scanner class and store them in variables.
2. To display information to the user using print and println.
3. To perform arithmetic calculations using arithmetic operations.

## Background

### 1. Display information to the user:

- Java provides the following two operations (actually there is one more, but we only focus on the following two at the moment) to display on monitor's screen.
  - i. `System.out.print()`
  - ii. `System.out.println()`
- To use these operations, you can follow the syntax below

#### Syntax:

```
// Print data on screen and the cursor will stay at the end of the string printed
System.out.print(<data> [+ <data>]);
// Print the data on screen and move the cursor to the beginning of the next line
System.out.println(<data> [+ <data>]);
// Print nothing
System.out.print("");
// Print nothing, but move the cursor to the beginning of the next line
System.out.println("");
```

- **Example**

```
public class OutputDataDemonstration
{
    public static void main(String[] args)
    {
        byte val1 = 10;
        short val2 = 412;
        int val3 = 58123;
        long val4 = 313123152;
        System.out.println("val1 & val2: " + val1 + " " + val2);
        System.out.println("val3 & val4: " + val3 + " " + val4);
        System.out.println();
        float val5 = 1.2345f;
        double val6 = 2.331256451313;
        System.out.println("val5 & val6: " + val5 + " " + val6);
        System.out.println();
        char val7 = 'D';
        System.out.println("Character in val7 is " + val7);
        System.out.println();
        boolean val8 = true;
        System.out.println("Boolean value in val8 is " + val8);
        System.out.println();
        System.out.print("Line A ");
        System.out.println("Continue to display data on line A");
```

## 2. Obtain input from user and store them in variables:

- Java provides a class named “Scanner” for getting user’s input from various input device, e.g. keyboard.
- The Scanner class is part of the package java.util. A package in Java is actually a collection of classes with a certain name. In this example, java.util is a package with so many classes inside, one of them is Scanner class.
- **Syntax:**

```
// Get the class definition of the Scanner class from java.util package
import java.util.Scanner;
// Create a Scanner object with name <object name>
// System.in is the “standard” input stream, which is keyboard
Scanner <object name> = new Scanner(System.in);
// Use the Scanner class’s operations (methods) to read input
<variable name> = <object name>.<method name>();
```

- A brief summary of Scanner class operations:

Method name	Description
boolean nextBoolean()	Returns the next input token as a boolean value
byte nextByte()	Returns the next input token as a byte value
short nextShort()	Returns the next input token as a short value
int nextInt()	Returns the next input token as a int value
long nextLong()	Returns the next input token as a long value
float nextFloat()	Returns the next input token as a float value
double nextDouble()	Returns the next input token as a double value
String next()	Returns the next input token as a String value
String nextLine()	Returns all input remaining on the current line as a String value

- **Example**

```
import java.util.Scanner;
public class InputDataDemonstration
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Please enter an integer: ");
        int intVal = sc.nextInt();
        System.out.println("The value you entered is " + intVal);
    }
}
```

### 3. *Perform calculation using arithmetic operations:*

- Java provides five basic operations for arithmetic calculation.
  - i. Addition (The operator used is +)
  - ii. Subtraction (The operator used is -)
  - iii. Multiplication (The operator used is \*)
  - iv. Division (The operator used is /)
  - v. Remainder (The operator used is %)

- **Syntax:**

*// Add the two values stored in variable 1 and variable 2*

*<name of variable for the result> = <name of variable 1> + <name of variable 2>;*

*// Subtract the value stored in variable 2 from variable 1*

*<name of variable for the result> = <name of variable 1> - <name of variable 2>;*

*// Multiply the value stored in variable 1 by the value stored in variable 2*

*<name of variable for the result> = <name of variable 1> \* <name of variable 2>;*

*// Divide the value stored in variable 1 by the value stored in variable 2*

*<name of variable for the result> = <name of variable 1> / <name of variable 2>;*

*// Find the remainder of value stored in variable 1 divided by the value stored in  
// variable 2*

*<name of variable for the result> = <name of variable 1> % <name of variable 2>;*

- **Example:**

```
public class ArithmeticOperationsDemonstration
{
    public static void main(String[] args)
    {
        int val1 = 21;
        int val2 = 10;
        int sum = val1 + val2;
        int diff = val1 - val2;
        int prod = val1 * val2;
        int quot = val1 / val2;
        int rem = val1 % val2;

        System.out.println("Addition: " + sum);
        System.out.println("Subtraction: " + diff);
        System.out.println("Multiplication: " + prod);
        System.out.println("Division: " + quot);
        System.out.println("Remainder: " + rem);
    }
}
```

## Lab Exercise

Write a Java program **SeparateDigits.java** that inputs an integer consisting of FIVE digits from the user, separate the input integer into its individual digits and prints the digits separated from one another by three spaces each. For example, if the user types in 12345, the program should print “1 2 3 4 5”. Assume that the user enters the correct number of digits.

(Note: The underlined text refers to the user input. Also, please follow closely to the output format shown below.)

Sample Output:

```
Enter a five digit integer: 12345
Digits in 12345 are 1 2 3 4 5
```

### Problem-solving tips

1. The input data consists of one integer, so you will use an int variable to represent it. Note that the description indicates that one five-digit number is to be input – not five separate digits.
2. You will use a series of statements to “break-down” the number into its individual digits, using integer arithmetic with remainder (%) and division (/) calculations.
3. After the number has been input, divide the number by 10000 to get the first digit. Why does this operation work? In Java, dividing an integer by an integer yields an integer result. Because the number input is five digits long, dividing it by 10000 gives the leftmost digit. For example,  $42339 / 10000$  evaluates to 4 because 10000 divides into 42339 four times. The remainder is truncated in integer arithmetic.
4. Change the number to a four-digit number, using the remainder operator to obtain the remainder after the number is divided by 10000 – in this case, the rightmost four digits. For example  $42339 \% 10000$  results in 2239.
5. Repeat this pattern of division and remainder calculations. Each time, the number used in the division and remainder calculations is reduced by a factor of 10. The first digit is obtained by dividing the give digit number by 10000. Then, the variable containing the number is assigned the remainder after the five-digit number is divided by 10000. After the number is changed to a four-digit number, perform division and remainder, calculations with 1000; after the number is changed to a three-digit number, perform division and remainder calculations with 100; and so on.

### **Questions:**

- Is there any limitation in your problem? Can you solve it?
- Is there any enhancement you can add to this program?

## Marking Scheme

The marking of this exercise will be based on the following criteria.

Graded items	Weighting
1. Correctness of program (i.e. whether your code is implemented in a way according to the requirements as specified.)	50%
2. Presentation of the Java codes (i.e. whether the program is properly indented, how close you follow the common conventions as mentioned in class, etc.)	30%
3. Documentation (with reasonable amount of comments embedded in the code to enhance the readability)	20%
	100%

**Be aware of plagiarism! DON'T copy the program file from your friends or classmates. If any identical copy is found, 5% of the coursework marks will be deducted for each of the file owner.**

## Program Submission Checklist

Before submitting your work, please check the following items to see you have done a decent job.

Items to be checked	☑ / ☒
1. Did I put my name, student ID and lab section at the beginning of all the source files as comment?	<input type="checkbox"/>
2. Did I put reasonable amount of comments to describe my program?	<input type="checkbox"/>
3. Are they all in .java extension and named according to the specification?	<input type="checkbox"/>
4. Have I checked that all the submitted code are compliable and run without any errors?	<input type="checkbox"/>
5. Did I zip my source files using Winzip / zip provided by Microsoft Windows? Also, did I check the zip file and see if it could be opened? <i>(Only applicable if the work has to be submitted in zip format.)</i>	<input type="checkbox"/>
6. Did I submit my lab assignment to Canvas before the deadline?	<input type="checkbox"/>