COMP1102/8702 – Practical Class 2

Variables, Assignment, Expressions and I/O

Aims and Objectives

This laboratory has been designed to help you to

- make use of variables to store, retrieve and update values,
- construct arithmetic expressions which make use of variables, literal values and operators such as integer "division" (/) and "remainder" (%),
- to create and use an input stream to read lines of text from the keyboard,
- convert a string of digits to an `int` value.

Getting Started

By default Java and *IntelliJ* expects the name of the class containing the main method to be the same as the name of the file containing the class. To avoid problems it is advisable to adhere to this convention. For example,

- Class name: `MyClass`
- Source file name: `MyClass.java`
- Class file (containing bytecode generated by the compiler): `MyClass.class`

On with the prac. . .

1. Create a directory called prac2 (in your cp1 directory) and make it your current working directory.
2. Copy the project directory ("Practical02") contained in the Week 2 area on FLO to your current working directory (prac2).

Refer back to practical 1 for information on using *IntelliJ* and running Java applications.

Task 1: Using Variables

In *IntelliJ* Open the project you have just copied. Open the file named `Assignment.java` and amend it so that it solves the following problem:

1. Declare an `int` variable called `total` with an initial value of 0
2. Declare an `int` variable called `val1` which has no initial value
3. Declare an `int` variable called `val2` which has no initial value
4. Assign 26 to the variable `val1`
5. Assign a value 1 bigger than the value of `val1` to the variable `val2` (you should not use the literal value 27).
6. Assign the sum of `val1` and `val2` to the variable `total`
7. Print out the values of `val1`, `val2` and `total` so that each is separated by a comma

Once you have successfully run your program your output should be similar to the following:

```
26, 27, 53
```

Checkpoint 7

Have the program source code and output marked by a demonstrator

Task 2: Reading, Storing and Outputting Strings

Open the file Echo.java (it should have been copied earlier) from *IntelliJ* and then compile and run it. It reads a line of input and echoes it back to the screen.

Modify the program so that it prompts for, and reads, two separate lines of input and outputs the contents of the lines in reverse order, on the same line, separated by a comma. Here is how your modified program should behave (user input is in **bold**):

```
Enter a line:
first
Enter another line:
second
second, first
```

Checkpoint 8

Have the program source code and output marked by a demonstrator

Task 3: String to Integer Conversion

Open the file Number.java - this is exactly the same as Echo.java

Modify the program so that rather than printing out the two strings which have been read in, it treats them as being two halves of a single integer. Your program should print out the value of that integer plus 1.

Hint: make use of the Integer.parseInt() method.

For example, after the following code has been executed the value of *k* will be the integer 44.

```
String s = "45" ;
int k = Integer.parseInt(s) - 1 ;
```

Here is how your modified program should behave (user input is in **bold**):

```
Enter a line:
23
Enter another line:
99
Number = 2400
```

The complete number is 2399 and adding 1 gives 2400.

Checkpoint 9

Have the program source code and output marked by a demonstrator

Task 4: Integer and Floating Point Division

Open Division.java and write code to solve the following problem:

1. prompts for and reads in an integer (on the same line)
2. Outputs:

i) the value of the number divided by 100 as a floating point value

ii) the remainder when the number is divided by 100

iii) the number of times 100 divides the integer

iv) outputs the digits of the integer **in reverse order**. That is, each digit must be extracted arithmetically.

You may assume the number is less than 10000.

Here are some examples of how your program should behave (user input is in **bold**):

```
Example 1
  Enter an integer: 1534
  15.34
  34
  15
  4351

Example 2
  Enter an integer: 403
  4.03
  3
  4
  304

Example 3
  Enter an integer: 4
  0.04
  4
  0
  4
```

Hint: `1534 % 10` has a value of `4` and `1534 / 10` has a value of `153`.

——————— **Checkpoint 10** ———————

Have the program source code and output marked by a demonstrator

Task 5 (Extension Practice): Variables and Division Expressions

Your task is to develop software for a machine which dispenses change. It accepts an amount in cents and outputs the ways that amount can be made up from 1, 5 and 20 cent coins (ignoring the fact that we no longer have 1 cent coins) using as many of the larger valued coins as possible.

Here is how your program should behave (user input is in **bold**):

```
Enter an amount of cents in the range 0 to 100: 77
Dispensing...
  3 20c coin(s)
  3 5c coin(s)
  2 1c coin(s)
```

Hint: `77 % 20` has a value of `17` and `77 / 20` has a value of `3`.