Laboratory 2 – Week 3

Java Variables

### 2.1 Introduction

**Firstly, this worksheet *is* one of the worksheets from which your laboratory worksheets portfolio of work will be assessed [CodeRunner tests].**

This laboratory session is designed to give you an understanding of the workings of Java variables. Remember if you want to keep any of your Java programs then you must save them on a network drive and not on the local computer’s hard drive.

Before you start this worksheet, we would like you to take the following survey:

<http://coderunner.brunel.ac.uk/>

Follow the link and complete the following steps:

1. Click on “CS1702 Introductory Programming (2018/2019)”.
2. Logon on with your usual computer ID.
3. Click on “1.0 - A Questionnaire on Your Programming Skills” under “Week 3”.
4. Please carefully read the information sheet.
5. If you are happy to complete the survey select – “Answer the questions...”.

Once you have completed the survey (or not…) carry on with the rest of this worksheet…

### 2.2 Assignment and Expressions

As we already know, variables are symbols whose value can change during the execution of a program. Create a new project and call it CS1702\_Lab2. Enter the following program:

**public** **static** **void** main(String args[])

{

**long** x;

x = 100;

System.*out*.println(x);

x = x + 10; //Line A

System.*out*.println(x);

x = x + 20; //Line B

System.*out*.println(x);

}

Run the program and note the results. What could you replace the code commented with Line A and Line B with in order to make the program mean the same and produce the same results?

Now modify the program as follows:

**public** **static** **void** main(String args[])

{

**long** x;

x = 100.1;

System.*out*.println(x);

x = x + 10.2; //Line A

System.*out*.println(x);

x = x + 20.3; //Line B

System.*out*.println(x);

}

Run the program and note the results. You should have found that Eclipse will not run the program and marks the modified lines down as being in error. Look at the *Problems* pane, what is the reason for the error? What happens when you change the type of x to double?

Note that we indent using a tab all of the program statements that lie between the class declaration and its end, and also then indent all the statements that lie between the declaration of the main entry point and its end. This enables us to easily understand the structure of the program we have written. The programs we are writing today may seem quite simple, but later we will write more and more complex programs, and thus correct indentation will be vital.

Often we want to copy text from the *console* Window, for example, we might want to save some results we have displayed from a program for later analysis. We can select the text in the *console* Window as we would in any other Windows application. We can then copy the text by right clicking the mouse over the selected text and choosing *Copy*. Try pasting the text into *Notepad* to check that the process works.

Write a new program that declares the following variables:

a = 100, b = 2.3, c = -52.2, d = true, e = "I am ",

f = "a student", g = 0, h = '!'.

Declare the following constants, pi = 3.142 and name = <your name>. Make sure you use the correct types. Declare three more variables, x as a long, y as a double and z as a String. Where possible use the expression shorthand notation. Where there are two expressions per exercise, these should be completed sequentially within the program. You may use as many lines for the expressions as needed.

Compute and display the following expressions, determine which ones are valid and work out why the invalid expressions are wrong. Verify your results using the Windows *Calculator*. Add **comments** where appropriate to remind you of what you have done and what your program does.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1) | y = a +b |  | 11) | x = (a +b)/(d+c) |
| 2) | y = a + d |  | 12) | y = 100.3  y = (y / (a + b)) - c |
| 3) | z = e + f |  | 13) | x2 = y2 + z2 |
| 4) | y = bc |  | 14) | name = "Is " + f - h |
| 5) | pi = 22/7 |  | 15) |  |
| 6) | z = name / g |  | 16) |  |
| 7) | z = c / g |  | 17) | y = b/g  z = y + name |
| 8) | x = 10  x = x3 |  | 18) | z = name  z = b/g + z |
| 9) | z = name + " is " + f + h |  | 19) | y = -2.3  y = ay2+by+c |
| 10) | y = (name + 10)/(name + h) |  | 20) |  |

To raise a variable a to the power of b, you might find the function Math.pow(a,b) useful.

### 2.3 Reserved Words

As we know, a **Reserved Word** is a special computer programming instruction which cannot be used as a variable name. For example, the symbol if is a reserved word within Java (and almost all other programming languages). Try and declare the following variables, determine whether they are reserved words or if they are invalid variable names:

for, four, 4, x4, x\_4b, while, While, switch, Switch, $witch, switch23, package, throw, new, name, friends, class, object, student, day, date, time, month, year, age, height, weight, 17beans, beans17, note, x\_\_\_

**2.4 Using Variables With Your Finch**

After the last printing line of your program, add the following code:

Finch myfinch = **new** Finch();

myfinch.setLED(255, 255, 255, 10000);

myfinch.quit();

In order for Eclipse to know where to find the Finch controlling code we are going to use, we have to add the following line before any class declaration:

**import** edu.cmu.ri.createlab.terk.robot.finch.Finch;

We will look into what an import is in a later lecture.

The meaning of each line of the program code is as follows:



The parameters for the setLED command (also called a method) are defined as follows:

|  |  |
| --- | --- |
| First Parameter: | Red level between 0 and 255 (of type int) |
| Second Parameter: | Green level between 0 and 255 (of type int) |
| Third Parameter: | Blue level between 0 and 255 (of type int) |
| Fourth Parameter: | Duration in milliseconds (of type int) |

Replace the four parameters with four variables named red, green, blue and duration. Make sure you declare them, assign values to them and then replace the numeric parameters in the call to the setLED method with the appropriate variables. Experiment with different values of the four variables. What happens if a variable is out of range, e.g. a negative number?

How would you change your program so that for a given variable red, green is twice red, blue is half way between red and green and duration is the sum of red, green and blue?

### 2.5 Further Reading

Visit the following web page <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/> and read up on the following topics:

1) Primitive Data Types

2) Variable Naming

3) Operators and Expressions