Dr Tomasz Kurowski
t.j.kurowski@cranfield.ac.uk

Practical Session 02
Control Statements, Loops, Methods

Part I: Control Statements

1. The front tyres of a car should both have the same pressure. The rear tyres of a car should also have the same pressure (but not necessarily the same pressure as the front tyres.) Write a program that reads in the pressure of the four tyres from the user, and writes a message that says if the inflation is OK or not. Use appropriate dialog boxes for user input / output.
2. Tyres don't have to have exactly the same pressure. Modify the program so that the front tyres can be within 3 psi of each other, and the rear tyres can also be within 3 psi of each other.

Part II: Loops

3. Create a new version of the very first exercise that displayed "Welcome to Java". Make this version display the message a user-defined number of times, where the user enters the number of iterations through an input dialog box. The output to the console should look something like

Welcome to Java for the 1 time

Welcome to Java for the 2 time

….

Welcome to Java for the N time

4. Modify this program to improve its grammar. Display the text in this format

Welcome to Java for the 1st time

Welcome to Java for the 2nd time

Welcome to Java for the 3rd time

….

Welcome to Java for the Nth time

Hint – you can use a collection of if-then-else statements, or a switch(), case construct. Limit N to 99.

5. Write a program that asks the user to enter a word. The program will then print this word for as many times as it has characters To do this you will need to use the String classes' length() method that counts the number of characters in a string

```
String inputString;
int    times;
     . . . .
times = inputString.length();
```

6. The Java Math class includes a method to create a pseudo random number between 0.0 and 1.0, of type double. (Note, a value of 1.0 will never actually be reached). A typical; call would be

```
double  dr = Math.random ();          // Returns a double in range 0.0 -> 0.99999…
```

Dr Tomasz Kurowski
t.j.kurowski@cranfield.ac.uk

Use the random number generator to create a pseudo random representation of a sequence of DNA, of length specified by the user. Use the capital letters C G A and T to represent the nucleotides. Use dialog boxes to accept the sequence length from the user, and to display the output.

7. A drug looses a certain percentage of its effectiveness every month it is in storage. When its effectiveness is below 50% it is considered expired and must be discarded. Write a program that accepts the % drop each month from the user, and determines how many months the drug can remain in storage. Display the results something like this (4% drop per month used in this case):

```
month: 0    effectiveness: 100.0
month: 1    effectiveness: 96.0
month: 2    effectiveness: 92.16
month: 3    effectiveness: 88.4735
month: 4    effectiveness: 84.9346
 .....                 ……
month: 17   effectiveness: 49.95868 DISCARDED
```

8. The Fibonacci series is the series of numbers 1 1 2 3 5 8 13 ....
Each number (except for the first two) is the sum of the previous two numbers. The first two numbers are 1 and 1.
Write a program to display 'n' elements of the Fibonacci series where 'n' is an integer value entered by the user.

9. Use the Newton Raphson iterative method to calculate the square root of a user-entered floating point value (Num). The Newton Raphson method allows you to approximate the square root of a number by repeatedly executing the following formula:

nextGuess = (lastGuess + (Num / lastGuess)) / 2

where Num is the original value. You should execute this loop until the difference between nextGuess and lastGuess is less than the required precision – say 0.00001. A good initial value of lastGuess might be Num / 2. You might also output the number of iterations taken to achieve the required precision.

HINT : This will be best implemented using a do { } while structure, where the terminating while condition will look something like

while (Math.abs(difference) > 0.00001);

(Math.abs( ) is a method to return the absolute value of the difference)

Dr Tomasz Kurowski
t.j.kurowski@cranfield.ac.uk

Part III: Methods

10. Write a method that converts Celsius to Fahrenheit using the following declaration:

```
public static double celsToFahr(double cels){ ...
}
```

The formula for the conversion is as follows:

*DegF = (DegC * 9 / 5) + 32.*

11. Write a program that uses a "for" loop and calls `celsToFahr` method in order to produce the following output:

```
Cels. Temp.        Fahr. Temp.
-------------------------------
40.0               104.0
39.0               102.2
38.0               100.4
37.0               98.60
36.0               96.8
35.0               95.0
34.0               93.2
33.0               91.4
32.0               89.6
31.0               87.80
```

12. Write a method `compoundInterest ()` that returns the value of a sum of money M after a number of time intervals Y, compounded at an interest rate R. So a call to the method would look something like

    newVal = compoundInterest (10000, 3, 7.5);

    All parameters should be of type double, and a double result returned. The first parameter should be the principle amount (e.g £10000), the second parameter the time period (e.g. 3 years) and the third parameter the interest rate (e.g 7.5% per annum).

    Hint : After one year, the new amount M would be M * (1 + R / 100)