Tutorial 2

Charlie Abela
Department of Artificial Intelligence
charlie.abela@um.edu.mt
Tutorial 2

In this tutorial we will focus on the control flow aspects of the Java language. The exercises are geared towards the use of if and if..else statements, switch and the various loops: for, while and do..while.

Problems

1. Write a program, RandomEven.java, that generates a random number between 0 and 100. If the generated number is even print out: “Number x is even” otherwise print out: “Number x is odd”.

2. Write a program, Exchange.java, that presents to the user two options, either to convert Lm into Euros or else to convert Euros into Lm. The user will then be allowed to enter an amount in respective currency and the output will be its equivalent in the other currency. Example of output when changing Euros to Lm:

   Enter amount in Euros:  12.50
   Exchange rate:  0.429
   Equivalent amount in Lm:  5.36

3. Write a program LogicGates.java, that presents the user with a choice between three logic gates: AND, OR and NOT. If the user chooses either AND or OR, then he will be allowed to enter two boolean values (from True or False) and the output will be either True or False depending on the gate and inputs. However should the choice be NOT, then, only one input will need to be entered. The output is the result from a NOT gate. Assuming that the user has chosen an AND gate a similar output will then be displayed:

   Using an AND gate

   Inputs
   A   B   Output
   True False False
4. Write a program, `LogicGateValidator.java`, which is based on the program `LogicGates.java`. However apart from choosing the gate and entering the inputs, the program now also accepts an output result. The program will verify whether the result is correct and outputs the appropriate message to the user. See example below:

Using an AND gate

<table>
<thead>
<tr>
<th>Inputs</th>
<th>A</th>
<th>B</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>False</td>
<td></td>
<td>False</td>
</tr>
</tbody>
</table>

Result: CORRECT

5. Write a program `ShapeArea.java`, which presents the user with the possibility of choosing any of three shapes: Rectangle, Triangle and Circle. After a choice is made the appropriate number of required inputs are elicited from the user and the area of the shape displayed.

6. Write a program, `VRTSimulation.java`, which allows the user to enter the details of a particular car (e.g. owner’s name, car model, year of manufacture etc). The program will then simulate a VRT process by generating random values between 0, 1 and 2 that respectively represent:
   i. 0: very bad condition (implies fail test and scrap),
   ii. 1: bad condition (implies fail test, repair and retake)
   iii. 2: good condition (implies pass test)

These values will be generated in respect of the following car features:
   a. brakes
   b. lights
   c. handbrake
   d. exhaust

7. Rewrite the above programs such that
   a. `RandomEven.java` allows the user to enter a number, representing the number of even integers to print. Call the new class `RandomEven2.java`.
   b. `LogicGates.java`, allows the user to iteratively choose different gates and inputs, and presents the appropriate output. Program needs also to display a way to exit, e.g. using a flag. Label the new class `LogicGates2.java`.
   c. Apply similar concept to `LogicGateValidator.java` and call it `LogicGateValidator2.java`. 


8. Write a program, \textit{VRTTest.java}, that given a year of car manufacture and a present year, identifies whether or not a VRT is due. VRT tests are performed every two years on cars that are more than 5 years old.

9. Write a program, \textit{HiLo.java}, whose objective is that of allowing the user to guess a program-generated secret number in the least number of tries. The secret number is an integer between 1 and 100, both inclusive. When the user makes a guess, the program replies with either Hi or Lo depending on whether the guess is higher or lower than the secret number. The maximum number of tries allowed is six, however, the user can play as many times as he wants, until some exit condition is satisfied.