| | |
|---|---|
| Module | CS4012 – Representation and Modelling |
| Day | 4 |
| Lab | 4 |
| Topic | Selection: the 'if-then' statement |

**Summary**

**(Statement):** A combination of one or more expressions that form one complete unit of execution.

As a program runs, it executes each of its statements one after another. However, a program with only one flow of execution, never altering in response to change, is limited. For this reason, Lua provides a set of control structures that allow for the flow of execution to be altered. The following examines the most basic of all control structures; namely, the "if-then" statement.

**Exercise 1:** The "if-then" statement

The "if-then" statement is the most basic form of control flow statement. When encountered, it tells your program to execute the code contained within its block only if some condition evaluates to true.

```
if condition then
        - - statement
        - - statement
end
```

**Task:**

1. Write a program that prints a message stating whether one number is greater than another.

Start with the code:

```
x = 7
y = 1

if(condition) then
        - - statement
        - - statement
end
```

The output should look something like:

"x is 7 and y is 1
x is greater than y"

2. Extend the above program to include a message that states whether the number is less than, or equal to another. Try different number combinations in the condition expression.

**Hint:** Use multiple "if-then" statements. And for the condition consider relational operators.

**Exercise 2:** The “if-then-else” statement

When an "if-then" condition evaluates to false the flow of execution jumps the “if-then” block and continues to the next line of code. The “if-then-else” statement provides a secondary path of execution.

```
if condition then
        - - statement
else
        - - statement
end
```

**Task:**

Write a program that prints a message stating whether a number is odd or even. The output should look something like:

“The number 3 is odd”
Or
“The number 4 is even"

**Hint:** For the condition consider arithmetic and relational operators.

**Exercise 3:** The “if-then - elseif ” statement

Imagine a scenario where there is conveyer-belt on which newly made hats are carried to a labeller, which labels the hats small, medium or large. If the hat is less than or equal to small, it gets labelled small. In like manner, if the hat is less than or equal to medium it gets labelled medium and so forth. Now consider sequential “if-then” statements where the size check is the condition and the labelling is the enclosed statement. A hat that checks out as small, also checks out as medium and large. Three labels is two too many! We need another statement that only gets checked if the preceding “if-then” statement evaluates to false. To this end, Lua provides the “if-then - elseif ” statement.

```
 if condition then
        - - statement

elseif condition then
        - - statement
end
```

**Note:** Multiple “elseif-then” statements are allowed following a single “if-then”. If an “else” statement is used, it follows the final “elseif-then” statement. Only one “end” is used in all case, and it appears at the end of the block.

**Task:**

A hat size can either be small, medium or large, where sizes are represented by number values. Write a program that determines the size a hat falls into based on whether it is smaller than or equal to one of the three standard sizes.