

Module	CS4012 – Representation and Modelling
Day	9
Lab	7
Topic	Repetition: the “for” statement

Summary

The “for” statement is a compact flow control structure used to iterate over a range of values. Distinct from the other looping constructs provided by Lua, it defines an explicit control variable as part of its declaration. Lua provides two forms of the range-based loop; the so-called generic and numeric “for” statements. The following focuses on the latter more flexible form.

Exercise 1: The numeric “for” statement

The “for” statement is typically used when the number of loop repetitions is known. The numeric form repeats a block of code for each number in an arithmetic progression. It has the following form:

```
for var = exp1, exp2, exp3 do
    -- statement(s)
end
```

The three expressions (exp1, exp2 and exp3) define the start, end and step size of the arithmetic progression, through which the variable “var” runs. Before the loop starts, the three expressions are evaluated. Note all expression must resolve to a number value. With each iteration of the loop the control variable “var” is assigned the current number in the progression. For example, the following loop repeats five times:

```
for start = 1, 5, 1 do          - -> initializes start to 1, and increases to 5, in steps of 1
    -- statement(s)
end
```

Tasks:

1. Write a program that uses a “for” statement to print the numbers 2 to 10 using a control variable that is incremented by 2.
2. Write a program that is a modification on the last; namely, the program counts down in steps of 2.
3. Write a program that uses a “for” statement to count up to 100 printing out numbers that are multiples of 5.
4. Write a program that uses a “for” statement to count up to 10 and outputs the first 10 multiples of 5. The output should look something like:

5, 10, 15, 20, 25, 30, 35, 40, 45, 50

Hint: The function “**io.write**”, unlike “**print**”, does not include the new line character (“\n”) by default, and therefore allows printing on the same line. The new line character can be added where appropriate.

Exercise 2: Nested “for” statements

A “for” statement can contain any statement within its code block including another “for” statement, which itself can contain nested statements. For instance, the following code snippet contains two “for” loops, one nested inside the other:

```
for i = 1, 5, 1 do
    for j = 1, 5, 1 do
        end
    end
end
```

The outer loop repeats five times and for each repetition, the inner loop repeats five times. Note also that the outer and inner control variables (“i” and “j”) have different names to avoid a name conflict. Consider what would happen if the inner control variable was the same as the outer variable – code it up run a test to see.

Task:

Write a program that uses nested “for” statements to create the following outputs:

a. * * * * *

 * * * * *

 * * * * *

 * * * * *

 * * * * *

b. *

 **

c. *

 **

Hint: This one is tricky. Two different patterns are repeated requiring more than one nested loop.

d. 1 2 3 4 5

 2 4 6 8 10

 3 6 9 12 15

 4 8 12 16 20

 5 10 15 20 25

Hint: Compare the last line and the output produced by task 4 in the previous exercise. Now consider if a changing control variable could replace a constant value.