Module	CS4012 – Representation and Modelling
Day	8
Lab	6
Topic	Repetition: the “while” and “repeat” statements

Summary

(Loop): A control structure that continually executes a code block while a particular condition is true.

Control structures act as junctions in a program’s code where the path of execution can be changed, causing certain blocks of code to be entered or simply passed-by entirely. A Loop is a class of control structure that allows the same block to be entered repeatedly. Lua provides three distinct loops for somewhat different use cases. The following considers loops in the form of the “while” and “repeat” statements.

Exercise 1: The “while” statement

The “while” statement is a flexible loop that is often used when the number of loop repetitions is unknown. When encountered it tells your program to repeatedly execute its block of statements as long as some condition expression evaluates to true.

```
while condition do
        - - statement
end
```

As the same block is repeatedly executed, it should contain a statement that controls the value of the condition expression; otherwise, an infinite loop can be created. For example, the following loop repeats five times:

```
count = 1           - ->  initializes count to the value of 1
while count <=  5  do     - - > checks that count is less than or equal to 5
        count = count +1           - ->  control statement, increments count by 1
end
```

Tasks:

1. Write a program that uses a “while” loop to print the values 1 to 10 using a count variable that is incremented by 1 at each loop.

2. Write a program that has a slight modification on the last: instead of counting up to 10 it counts down from 10.

3. Modify the previous program so that a different message is printed for odd an even numbers of the count variable.

Hint: Use an “if-then else” statement to distinguish between odd an even.

Exercise 2: The “repeat” statement

The “repeat” loop is most often used when a block of code should be executed at least once. The difference between the “while” and “repeat” loops is that “repeat” evaluates its expression at the bottom of the loop instead of the top.

```
repeat
        - - > statement
until condition
```

Similarly, the “repeat” block should contain a statement that controls the value of the condition expression. For example, the following loop repeats five times:

```
count = 5          - ->  initializes count to the value of 1
repeat
        count = count - 1          - ->  control statement, decrements count by 1
until  count <= 0
```

Tasks:

1. Write a program that uses a “repeat” loop to print the values 1 to 10 using a count variable that is incremented by 1 at each loop.

2. Write a program that uses a “repeat” loop to sum the numbers 1 through 10, printing the result of the summation.

Hint: You’ll need another variable to hold the sum.

Exercise 3: Baggage handling

At airport baggage handling checked bags circle on a carousel awaiting individual travellers to pick up their bag. Standing at the carousel, it is random when the correct bag will arrive but in every case a traveller must wait until the first bag arrives.

Task:

Start with the following code:

```
math.randomseed(os.time())
myBagNum = math.random(1,10)          --> randomly generates a value between 1 and 10
```

Write a program that uses a loop to simulate how many bags a traveller checks before finding their own. The output should look something like:

```
Bag 1: not mine.
Bag 2: not mine.
Bag 3: mine!
```

Hint: Compare the current loop count with the random bag value “myBagNum”.