Module	CS4012 – Representation and Modelling
Day	2
Lab	2
Topic	Variables and Data Types

Summary

(Variable): A name to store and identify a value.
(Type): An inherent property of a value that defines the operations that can be performed on it.

As a program runs, each line of code is executed sequentially, one after another. Variables allow data values defined in one line of a program's code to be used in the same or in another line. In turn, the operations that can be performed on a value are determined by its data type. There are eight basic types in Lua; but, for the purpose of this lab session we focus on five: nil, boolean, number, string and function. The following focuses on the step-by-step nature of a computer program and on the data types of literal values.

Exercise 1: creating an algorithm

An algorithm is a step-by-step set of operations performed to solve a problem – a program is an implementation of an algorithm. Pseudo code lies somewhere between your own words and a programming language (tending toward the former), and represents a way to sketch out an algorithm before jumping into true code.

For instance, the following represents an algorithm to make tea:

```
boil water
put teabag into pot
pour water to spout
if(weak)
        wait 3 minutes
if(strong)
        wait 2 minutes more
stir
remove tea bags          // comment otherwise turns to porter
```

Task:

Write in pseudo code an algorithm for making a pizza, using pen and paper. Think of the steps in the form of print statements and covert your step-by-step algorithm into a Lua program.

Hint: Maybe start by rolling the dough.

Exercise 2: assigning values to variables

By convention, a variable can be any sequence of letters, digits, and underscores, not beginning with a digit. A value is assigned to a variable using the assignment operator ("="), for example:

```
_stub1 = "My name is"
```

It is also possible to perform a multiple assignment in Lua, where a list of values is assigned to a list of variables in one step. For example, in the following assignment:

x, y = 10, “apple”

the variable “x” gets the value 10, and “y” gets the string literal “apple”

Tasks:

1. Write a program that prints your name and student number by concatenating variables holding these values. The output should look something like: “My name is Sam and my student number is 12345678”.

Hint: The string concatenation operator (“..”) is used to join two strings together. For example, the variable _stub1 can be joined (or concatenated) with the string “Sarah” as shown in bold:

myName = **_stub1 .. “ Sarah”** **- -> “**myName” is equal to the value “My name is Sarah”

2. Write a program that swaps two values using a multiple assignment described previously. The output should look something like:

“x was 10, y was apple
x is now apple, y is now 10”

Hint: Try assigning the variable “y” (on the left the “=” operator) to the variable “x” (on the right) using single assignment. Now extend this to the multiple assignment.

Exercise 3: finding the data types of values

The “type” function gives the type name of a given value as a string; the “print” function can output that name to the screen. Calling “type” is done in the same way as “print”.

Task:

Write a program that prints to the screen the type name of the following values:

a. “Hello World”
b. 12
c. 10.12
d. true
e. false
f. nil
g. print
h. type(print)

The output for the first line should look something like: “Type (“Hello World”) equals: string”.