Concept	Keyword(s)/Symbols	Example
Commenting		

Comment	`//`	`//This function squares a number`
		`function squared(number)`
		`  squared = number^2`
		`  return squared`
		`endfunction`
		`//End of function`

Variables		
Assignment	`=`	`x = 3` `name = "Louise"`
Constants	`const`	`const vat = 0.2`
Global Variables	`global`	`global userID = "Cust001"`
Input/Output		
Input	`input(…)`	`myName = input("Please enter a name")`
Output	`print(…)`	`print("My name is Noni")` `print(myArray[2,3])`
Casting		
Converting to another data type	`str()`	`str(345)`
	`int()`	`int("3")`
	`float()`	`float("4.52")`
	`real()`	`real("4.52")`
	`bool()`	`bool("True")`

Concept	Keyword(s)/Symbols	Example
Iteration		

FOR loop (Count-controlled)	`for … to …`
	`next …`
		`for i=0 to 9`
		`  print("Loop")`
		`next i`
		This will print the word "Loop" 10 times, i.e. 0-9 **inclusive**.

	`for … to … step …`
	`next …`
		`for i=2 to 10 step 2`
		`  print(i)`
		`next i`
		This will print the even numbers from 2 to 10 inclusive.
		
		`for i=10 to 0 step -1`
		`  print(i)`
		`next i`
		This will print the numbers from 10 to 0 inclusive, i.e. 10, 9, 8,…, 2, 1, 0.
		
		Note that the `'step'` command can be used to increment or decrement the loop by any positive or negative integer value.

WHILE loop (Condition-controlled)	`while …`
	`endwhile`
		`while answer != "Correct"`
		`  answer = input("New answer")`
		`endwhile`
		Will loop until the user inputs the string "Correct". Check condition is carried out before entering loop.

DO UNTIL loop (Condition-controlled)	`do`
	`until …`
		`do`
		`  answer = input("New answer")`
		`until answer == "Correct"`
		Will loop until the user inputs the string "Correct". Loop iterates once before a check is carried out.

Concept	Keyword(s)/Symbols	Example
Selection		

IF-THEN-ELSE	`if … then`
	`elseif … then`
	`else`
	`endif`
		`if answer == "Yes" then`
		`  print("Correct")`
		`elseif answer == "No" then`
		`  print("Wrong")`
		`else`
		`  print("Error")`
		`endif`

CASE SELECT or SWITCH	`switch … :`
	`  case … :`
	`  case … :`
	`  default:`
	`endswitch`
		`switch day :`
		`  case "Sat":`
		`  print("Saturday")`
		`  case "Sun":`
		`  print("Sunday")`
		`  default:`
		`  print("Weekday")`
		`endswitch`

Concept	Keyword(s)/Symbols	Example
String handling/operations		
String length	`.length`	`subject = "ComputerScience"` `subject.length` **gives** `the value 15`

Substrings	`.substring(x , i)`
	`.left(i)`
	`.right(i)`
		`subject.substring(3,5)` **returns** `"puter"`
		`subject.left(4)` **returns** `"Comp"`
		`subject.right(3)` **returns** `"nce"`
		x is starting index; i is number of characters; 0 indexed

Concatenation	`+`	`print(stringA + stringB)` `print("Hello, your name is: " + name)`
Uppercase	`.upper`	`subject.upper` **gives** `"COMPUTERSCIENCE"`
Lowercase	`.lower`	`subject.lower` **gives** `"computerscience"`
ASCII Conversion	`ASC(…)` `CHR(…)`	`ASC('A')` **returns** `65` (numerical) `CHR(97)` **returns** `'a'` (char)

Concept	Keyword(s)/Symbols	Example
File handling		
Open	`open(…)`	`myFile = open("sample.txt")` **Note that the file needs to be stored as a variable.**
Close	`.close()`	`myFile.close()`
Read line	`.readLine()`	`myFile.readLine()` **returns** the next line in the file
Write line	`.writeLine(…)`	`myFile.writeLine("Add new line")` **Note that the line will be written to the END of the file.**

End of file	`.endOfFile()`	`while NOT myFile.endOfFile()`
		`  print(myFile.readLine())`
		`endwhile`

Create a new file	`newFile()`	`newFile("myText.txt")` **Creates a new text file called "myText". The file would then need to be opened using the above command for Open.**
Arrays		

Declaration	`array colours[…]`	`array colours[5]`
		Creates 1D array with 5 elements (index 0 to 4).
		
		`array colours = ["Blue", "Pink", "Green", "Yellow", "Red"]`
		Arrays can be declared with values assigned.

Arrays are **0 indexed** Arrays only store a **single data type**	`array gameboard[…,…] = …`	`array gameboard[8,8]` **Creates 2D array with 8 elements (index 0 to 7).**
Assignment	`names[…] = …` `gameboard[…,…] = …`	`names[3] = "Noni"` `gameboard[1,0] = "Pawn"`

Concept	Keyword(s)/Symbols	Example
Sub programs		

Procedure	`procedure name(…)`
	
	`endprocedure`
		`procedure agePass()`
		`  print("You are old enough to ride")`
		`endprocedure`
		
		`procedure printName(name)`
		`  print(name)`
		`endprocedure`
		
		`procedure multiply(num1, num2)`
		`  print(num1 * num2)`
		`endprocedure`

Calling a procedure	`procedure(parameters)`	`agePass()`
		
		`printName(parameter)`
		
		`multiply(parameter1, parameter2)`

Function	`function name(…)`
	`  …`
	`  return …`
	`endfunction`
		`function squared(number)`
		`  squared = number^2`
		`  return squared`
		`endfunction`

Calling a function	`function(parameters)`	`print(squared(4))`
		
		`newValue = squared(4)`
		Note: Function returns should be stored in a variable if needed for later use in a program.

Random numbers		

Random numbers	`random(…,…)`	`myVariable = random(1,6)`
		Creates a random integer between 1 and 6 inclusive.
		
		`myVariable = random(-1.0,10.0)`
		Creates a random real number between -1.0 and 10.0 inclusive.