2.2 PROGRAMMING FUNDAMENTALS

DATA TYPES

Data Type	Definition
String	Text eg: “Hello”
Integer	Whole number eg: 32
Float/Real	Decimal number eg: 1.2
Boolean	Two values eg: true or false
Character	A single character eg: b

Casting is when you want to change between data types. Eg – if you want to use an integer in a sentence you would need to convert it to a string.

VARIABLES AND CONSTANTS

Variable – A value which may change while the program is running. Variables can be local or global.

Local Variable – a variable which can only be used within the structure they are declared in.

Global Variable – a variable which can be used in any part of the code after they are declared

Constant – A value which cannot be altered as the program is running.

OPERATORS

Operator/Function	Definition
Exponentiation	Raises a number to a power eg: 2**3 OR 2 ^3 (=2^3)
Quotient/DIV	Gives the whole number after a division
Remainder/MOD	Gives the remainder part of a division
==	Is equal to
! or <>	Is not equal to
<	Is less than
>	Is more than
>=	Is more than or equal to
<=	Is less than or equal to

ARRAYS

One-Dimensional Arrays- this is like a list. In this example an array has been created called students. The list can hold 3 items (as shown).

```
array students [3]
students [0] = “Bob”
students [1] = “Dave”
students [2] = “Bob”
```

This command would print the second item (1) From the array. It would print “Dave”.

```
print(students[1])
```

Two-Dimensional Arrays – these are lists within lists (like a table)

```
Grades=[[“Bob”, “22%”, “44%”], [“Dave”,
“85%”, “100%”]]
```

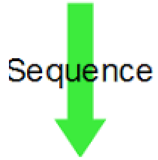
The code above creates the 2D array. The code Below would output:
“Bob’s first test score was 22%”

	0	1	2
0	Bob	22%	44%
1	Dave	85%	100%

```
print(“Bob’s first test score was “ + Grades [0, 1]
```

2.2 PROGRAMMING FUNDAMENTALS CONTINUED

PROGRAMMING CONSTRUCTS



A Sequence is when there are programming steps that are carried out one after another.

Selection is where there are different paths in your code eg: IF, ELIF, ELSE

Iteration is when there is repetition (loops) in code. This could be a WHILE loop (do something WHILE a condition is met) or a FOR loop (do something for a set number of times)

This count-controlled loop would print “Hello World” 8 times.:

```
for i=0 to 7
    print (“Hello”)
next i
```

These condition controlled loops would check if a password’s correct:

```
while answer != ”letmein123”
    answer=input(“Enter password”)
endwhile
```

```
do
    answer=input(“Enter password”)
until answer==”letmein123”
```

STRING MANIPULATION

0	1	2	3
W	o	r	d

The characters in a string are numbered starting with position 0.

Function	Purpose
x.length	Gives the length of the string
x.upper	Changes the characters in the string to upper case
x.lower	Changes the characters in the string to lower case
x[i]	Gives the character in position i. Eg: x[2] = “r”
x.substring(a,b)	Gives the characters from position a with length b. Eg: x.subString(1,2) = or
+	Joins (concatenates) two strings together

FILE HANDLING

Myfile=openRead(“myfile.text”)	Opens the file in read mode
Myfile=openWrite(“myfile.text”)	Opens the file in write mode
Myfile.writeLine (“Hello”)	Writes a line to the file
Line1=myfile.readLine()	Reads one line of the file
Myfile.close()	Closes the file
endOfFile()	Used to determined the end of a file

IF/ELSE AND SWITCH/CASE FOR SELECTION

Selection can be shown using IF/ELSE or SWITCH/CASE

IF ELSE	SWITCH/CASE

If choice == “a” then
   print(“You chose A”)
elseif choice==”b” then
   print(“You chose B”)
else
   print(“Unrecognised choice”)
	Switch entry:
	   case “A”:
	      print(“You chose A”)
	   case “B”:
	      print(“You chose B”)
	   default:
	      print(“Unrecognised choice”)

2.2 PROGRAMMING FUNDAMENTALS CONTINUED

SUB PROGRAMS

Procedures are a set of instructions stored under a name so that you can call the procedure to run the whole set of instructions.
A **function** is like a procedure but always returns a value.
Parameters are variables used to pass values into a function or procedure.

A procedure **with** parameters	A procedure **without** parameters

procedure intro (name)
    print(“Hello “ +name)
    print(“Welcome to the game”)
endprocedure
	procedure intro ()
	    print(“Hello”)
	    print(“Welcome to the game”)
	endprocedure

Functions must take at least one parameter and must return a value:

```
function double(number)
    print number*3
endfunction
```

SQL (Structured Query Language)

SQL is the language used to manage and search databases.

Commands	Example	What it does

SELECT
FROM
	SELECT name, age
	FROM students
		Displays the name and age of everyone in the students table

WHERE	SELECT name FROM students
	WHERE gender=male
		Displays the name of everyone in the students table who’s gender is male

LIKE	SELECT name FROM students
	WHERE name LIKE “% Smith”
		Displays the students names that end with Smith.

AND	SELECT name FROM students
	WHERE gender=male AND
	attendance > 90
		Displays the students who are male and have an attendance of more than 90.

*	SELECT * from students	Selects all of the fields from the students table

RECORDS

Records are a data structure used to store a collection of data. They can store information of different data types.
Field = each item in a record is a field. Each field has a name and data type.

A record can be created like this:

```
record students
    int student_number
    string student_name
    bool passed_test
endrecord
```

Data can be assigned using variables:

```
Student1=students(1,”Bob Jones”, True)
Student2=students(2,”Steve Smith”, False)
Student3=students(3,”Sally Roberts”, True)
```

The whole record can be accessed using the variable name:

```
print(Student1)
```

(1, “Bob Jones”, True)

or part of a record can be accessed:

```
print(Student3.student_name)
```

Sally Roberts