AS Computer Studies: PROGRAMMING

ITERATION 1

Starter

- Put the following code in order (write down the line numbers).
- The program should display the numbers 1-24 on screen.

1. Console.writeline(pos)
2. Pos = pos + 1
3. Pos = 1
4. Loop
5. Do while pos < 25
6. Dim pos as integer

Answer: 6, 3, 5, 1, 2, 4

Learning Objectives

- Understand what is meant by **iteration** in programming.
- Understand the **use of iteration** in programming.
- State when a loop will end.
- State when the **For … To … Next** loop should be used.
- State the general form of the **For … To … Next** loop.

Recap: What is Iteration?

- ***repetition*** of a sequence of (one or more) statements.
- In programming, this is often referred to as a ***loop***.
- Contains a **Boolean condition** that determines when it terminates.
- The statements might not be executed at all (***zero repetitions***), or may be executed at least once. Eventually, something **must** ***stop*** the repetition, allowing the program to continue further.

Types of loops:

- **For … To … Next**
- **Do While** (condition is true) … **Loop**
- **Do …. Loop Until** (condition is true)

Loop Body

- The code inside the loop (inserted in place of the dots between).

Why bother with iteration?

- Imagine a list (**array**) of 1000 entries, each containing a payment due from customers. At the end of the month, £20 bonus is added on to each customer. You would need to write at least 1000 lines of code to do this.
- You could just loop through the array and add it on in about 6 lines of code!
- Imagine counting the number of letter "e's" which appear in this slide. This would involve you going through the slide and checking each word, then adding 1 to a tally every time you come across one.
- Alternatively, a loop could go through the slide and do this for you.
- A loop is used when you are performing **repetitive tasks**.

Caution About Loops

- The distinction between
  - **checking** ***before starting*** **any processing**, and
  - **checking at the** ***end*** **of each run through the statements**

is a fundamental one.

- ***Putting the check in the wrong place is one of the commonest causes of errors when dealing with loops in programs.***
- For example:
  - Would you issue a bill to a customers **BEFORE** checking if they owed anything or would you check each customers balance before printing the bill.

For ….. To …. Next….

- Used when you know **exactly** how many times the code must be repeated.

e.g. Display the numbers from 1 to 10.

```
Dim Number As Integer
For Number = 1 To 10
    Console.Writeline(Number)
Next Number
Console.Readline()
```

General syntax:

```
For (variable_identifier = start_value) To (end_value)
        (Loop Body statements) …
Next (variable_identifier)
```

Note:

Start and **End** values may be **integer** *constants, variables or expressions*.

The *variable_identifier* in the last line of the loop is optional but it is good practice to include it as it makes your code easier to read.

For ….. To …. Next…. (dry-running the algorithm)

- The first time:

  For Number = 1 To 10

  is executed, Number is set to 1.
- The loop body code is then executed - the number one is displayed in on the console.
- The line:

  Next Number

  indicates the **end of the loop** and the variable **number** is incremented by 1; the program loops back to the first line:

  For Number = 1 To 10
- The loop body code is then executed again and this time the number two is displayed on the Console.
- This process continues **until the loop has been executed exactly 10 times**.

For ….. To …. Next…. (dry-running the algorithm)

Pink booklet

Page 30

Understanding code:

-table, 2nd example – let's 'dry-run' this algorithm.

Checking if the contents of a control / variable is numeric

- **IsNumeric**(….Text) -> if used with an object (tool from the Toolbox)

  Name of control

- **IsNumeric**(*variable_name*) -> if used with a simple variable.

- Returns **True** if numeric and **False** if it is not.

Multiplication table

Do this ....

Specification:

- Ask the user to enter a number
- Then output the multiplication table

for that number.

Start a new console application; name it multiplication table.

Multiplication table

Type in the code to ask the user for a number; read it into a variable you must declare!

Check that the typed number actually IS a number; if not, ask the user to type again (hint: use a DO Until loop?)

The following should appear in your code (not necessarily in this order):

```
Dim Number, Index, Result As Integer
…….
Number = Console.Readline()
For Index = 1 To 12 ‘Repeat the following from 1 to 12.
    Result = Index * Number
    Console.Writeline(Index & " x " & Number & " = " & Result)
Next Index
…….
```

Run the program and test it thoroughly.

What if you want to step in different values?

- What if you don't always want to go up in 1's?
- E.g. What if you wanted to print the odd numbers between 1-19. You would need to go up in 2's!
- To do this, you just add the STEP keyword into your FOR declaration.

```
For controlvar = startval To endval Step stepval
  statement(s)
Next controlvar
```

Try this example!

Today’s Task

Pink booklet:

– Task 14(p28-30) –complete the table on the booklet and do the programming questions

PLUS:

Write a program that:

- Asks the user for two different numbers.
- Shows all the numbers from the first value to the second value (given above).

Extension:

Show only numbers between the two values not the values themselves.

Stop the user entering letters (so check if the user has entered a number!).

Learning Objectives

- Understand what is meant by **iteration** in programming.
- Understand the **use of iteration** in programming.
- State when a loop will end.
- State when the **For … To … Next** loop should be used.
- State the general form of the **For … To … Next** loop.

Plenary

- What is a program loop?

Sections of code that may be repeatedly executed.

- How does the loop terminate?

Contains a Boolean condition that determines when it terminates.

- When should the For … To … Next loop be used?

Used when you know exactly how many times the code must be repeated.

Plenary

- What is the general form of the For … To … Next loop?

For (variable identifier = start value) To (end value)

(Loop Body statements) …

Next (variable identifier)

- What is the general form of code which will check if the contents of a control are numeric?

IsNumeric(control_name.Text)

- What is returned if the contents is numeric and what is returned if not?

True / False