GOVERNMENT OF
WESTERN AUSTRALIA

TAFE WA South Metropolitan

Flowcharts

(Session 13)

Review - Pseudocode Rules

1. Like regular code – **write one statement per line**
2. Use CAPITALISATION for keywords
3. Indentation
   - This shows hierarchy and will get you used to programming
4. End multiline structures
5. Keep statements language independent
   - This is not a program, its **<u>plain English</u>**

<u>Pseudocode</u>

```
READ name, gross_pay, taxes
IF taxes > 0
    net = gross_pay – taxes
ELSE
    net = gross_pay
ENDIF
WRITE name, n
```

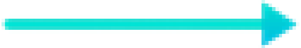
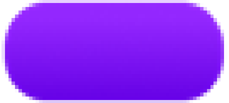
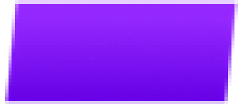
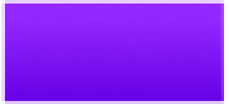
Overview

- Flowcharts
- Sequential Structure
- Selection Structure
- Loop Structure
- From Flowcharts to Python Code

Structures

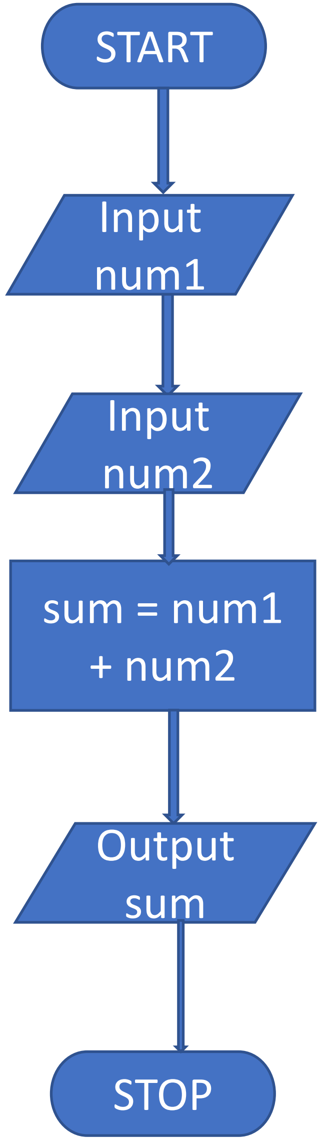
- Three basic types of control structures:
    - **sequential**
    - **selection**
    - **repetition**
- Throughout this section we are going to review the structures which you will use in pseudocode and your developed programs
- To assist in your understanding we will also use flow charts

Review (Flowcharts)

- Graphically represent the flow of your program
- Flow line 
- Terminal (start / stop) 
- Input/output – used for input output operations 
- Processing used for data manipulation 

Sequential

- **INPUT num1**
- **INPUT num2**
- **sum = num1 + num2**
- **WRITE sum**



START

Input num1

Input num2

sum = num1 + num2

Output sum

STOP

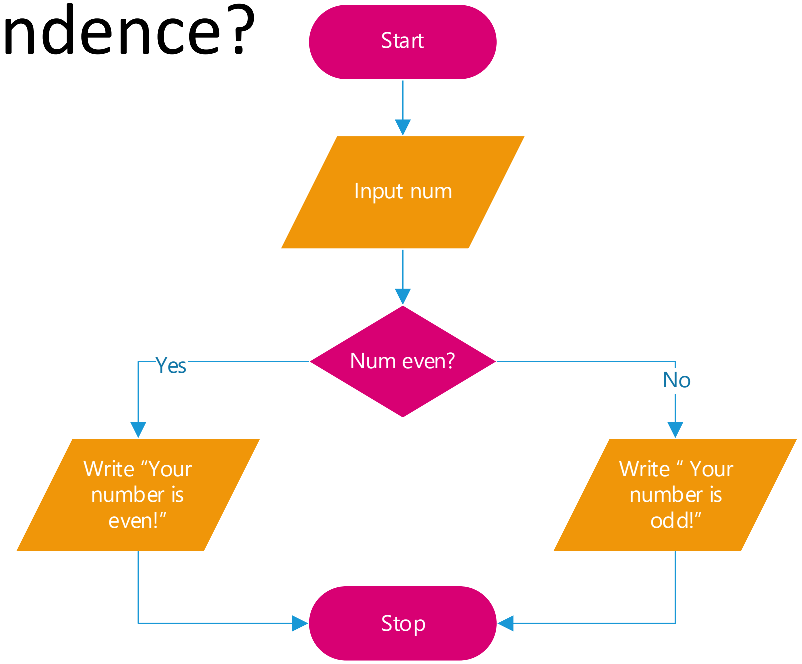
Selection Structure

- Selection is as simple as yes or no. If a user enters a number, is it even? If yes, it’s even, if no, it’s odd.
- In Pseudocode this looks something like:

```
INPUT num
IF num is even
    WRITE “Your number is even!”
ELSE
    WRITE “Your number is odd!”
ENDIF
```

Selection Structure

- If we look at the flow chart and the pseudocode – can you see the correspondence?

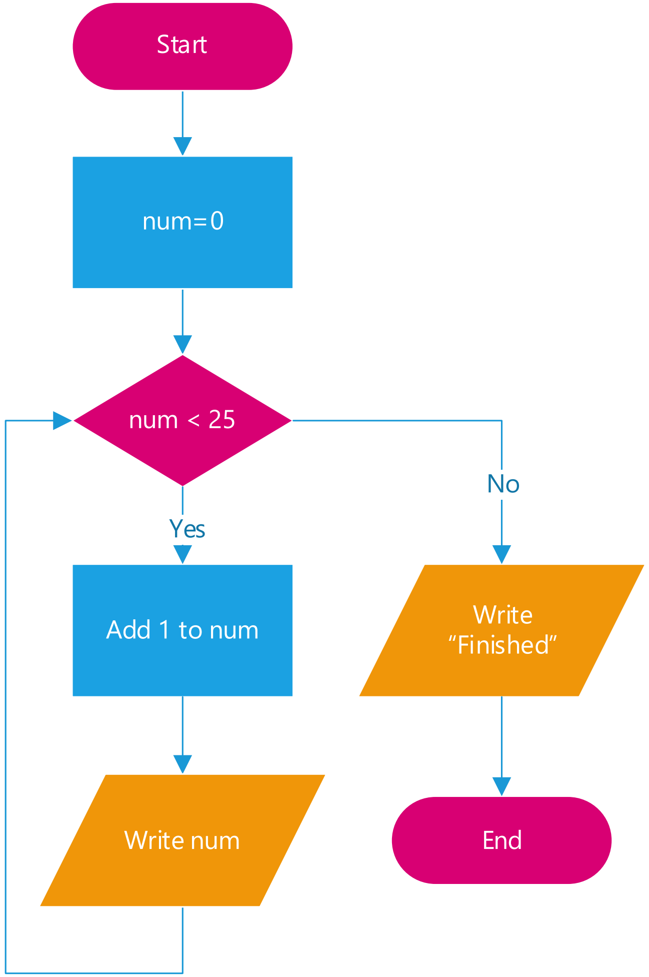


```
  INPUT num
  IF num is even
      WRITE "Your number is even!"
➔ ELSE
      WRITE "Your number is odd!"
  ENDIF
```

Looping Structure

- Loops can be a bit confusing especially when you attempt to depict looping in flow charts.
    - This is due to the use of the diamond as a control symbol
- Thankfully pseudocode does not require a control symbol, and we can use specific keywords such as WHILE/ENDWHILE and FOR/ENDFOR
- Lets take a look at both of these.

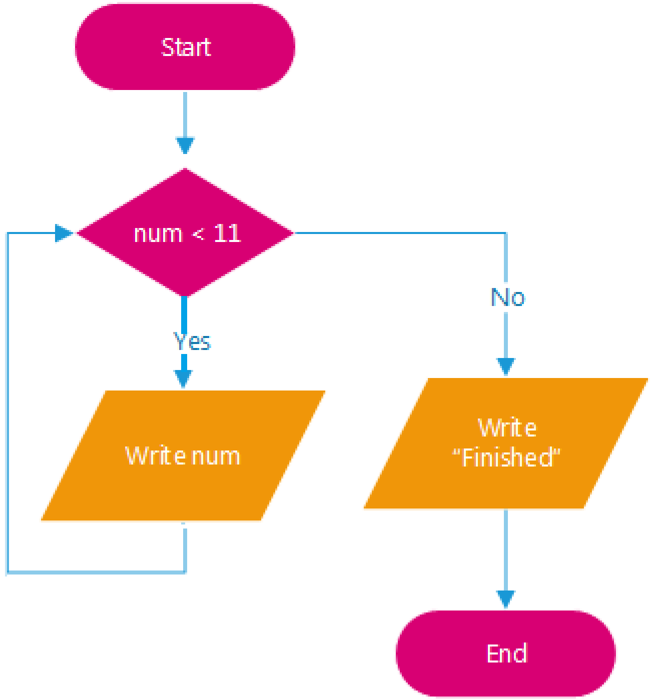
Looping Structure: WHILE/ENDWHILE



Pseudocode ➔

```
num = 0
WHILE num < 25
      ADD 1 to num
      WRITE num
ENDWHILE
WRITE "Finished"
```

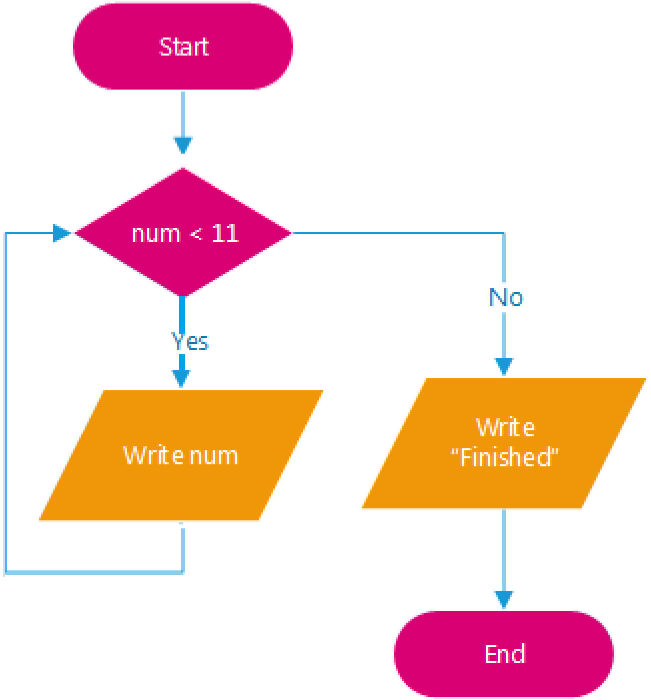
Looping Structure: FOR/ENDFOR



Pseudocode ➔

```
FOR num = 1 to 10
        WRITE num
ENDFOR
WRITE "Finished"
```

From Flowcharts to Python Code



Pseudocode

```
FOR num = 1 to 10
        WRITE num
ENDFOR
WRITE “Finished”
```

➔

Python Code

```
for num in range(1, 11):
    print(num)
print("Finished")
```

Comparing Pseudocode and Flowcharts

- Pseudocode Advantages:
    - Easy to modify
    - Structured concepts
    - No special software requirement to write it
- Pseudocode Disadvantages
    - No visuals (can be hard to follow)
    - No accepted standards (can vary between organisations)

Questions?