**Diploma in**

Computer Science

Formulating solutions

Explore pseudocode as away of representing algorithms

Appreciate flowcharts as an additional way of representing algorithms

Identify the symbols used in flowcharts

Explore problems and create pseudocode and flowcharts

Objectives

# Why is it called pseudocode?

Purpose of pseudocode

To show how an algorithm or proposed solution should work

What is pseudocode?

- Demonstrates where a particular mechanism or technique will be

- Statements that describe a concept

- Algorithm of informative text and annotations in English

…the origin of the word 'algorithm'?

Why do you need pseudocode?

- Industry practice

- Can't show a client cryptic lines of code

- Need something understandable

- More readable for people unfamiliar with code

Why do you need pseudocode?

- Enables you to express ideas and lay out your thoughts

- Best way to start implementing your algorithm

- Bridge the gap between the algorithm and the flowchart

Why use pseudocode?

- Spells out portions of code

- Serves as a reference

- Documentation of sorts: written text or illustrations that accompany software, explaining how it operates and all the information pertaining to the software development cycle
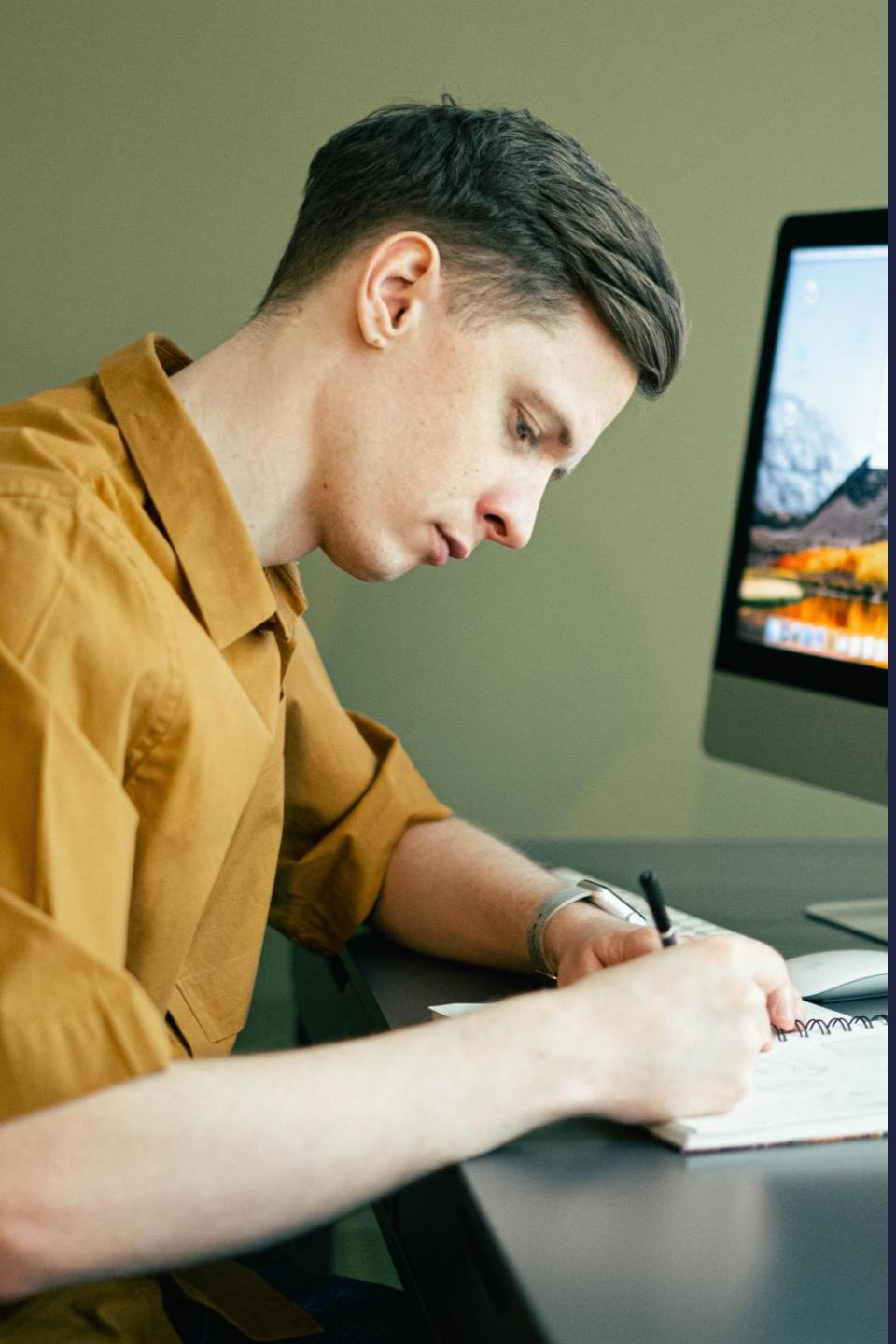
Goal of pseudocode

CLARITY

UNIVERSAL

Pseudocode: laying out your thoughts

- Every sentence must say just one thing.
- Pseudocode should not be too general.
- Clarity is all important.

Analyse the problem

Arrange the tasks in sequence

Start with a statement of the goal

General guidelines for writing pseudocode

- Use naming conventions when naming variables

  Example: number1 and number2

- Use indent to make loops and decisions easier to follow

- Use conventional programming structures

- Don't leave anything to assumption

Good practices when writing pseudocode

Blocks of code need to be indented to show a selection or iteration structure

Capitalise key commands

Organise pseudocode in sections

Common terms in pseudocode

Input: used when inputting something

For: used for a counting loop

Output: used when output appears on the screen
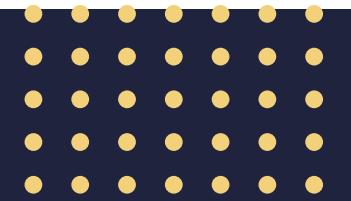
If-then-else: used to denote a choice or decision to be made

While: used to indicate a loop

Repeat-until: used when a block of code is repeated until a condition is met

Baking a cake
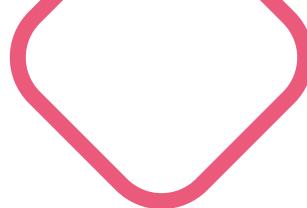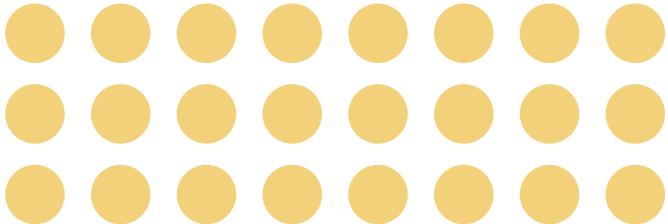
Pseudocode for baking a cake

```
This program bakes a cake
START
Get ingredients
PROCEDURE mix ingredients
        Get bowl
Add 2 cups flour
        Add 1 cup sugar
Add raisins
        Add cocoa
        Add eggs
        Add baking powder
        Turn on mixer
PROCEDURE preparation
        Grease pan
        Preheat oven
PROCEDURE bake
        Pour dough into pan
        Put in oven
        Bake 20 mins
                IF cake is ready
                THEN take out of oven
                Cool and eat
                ELSE add 5 minutes
                Bake
END
```
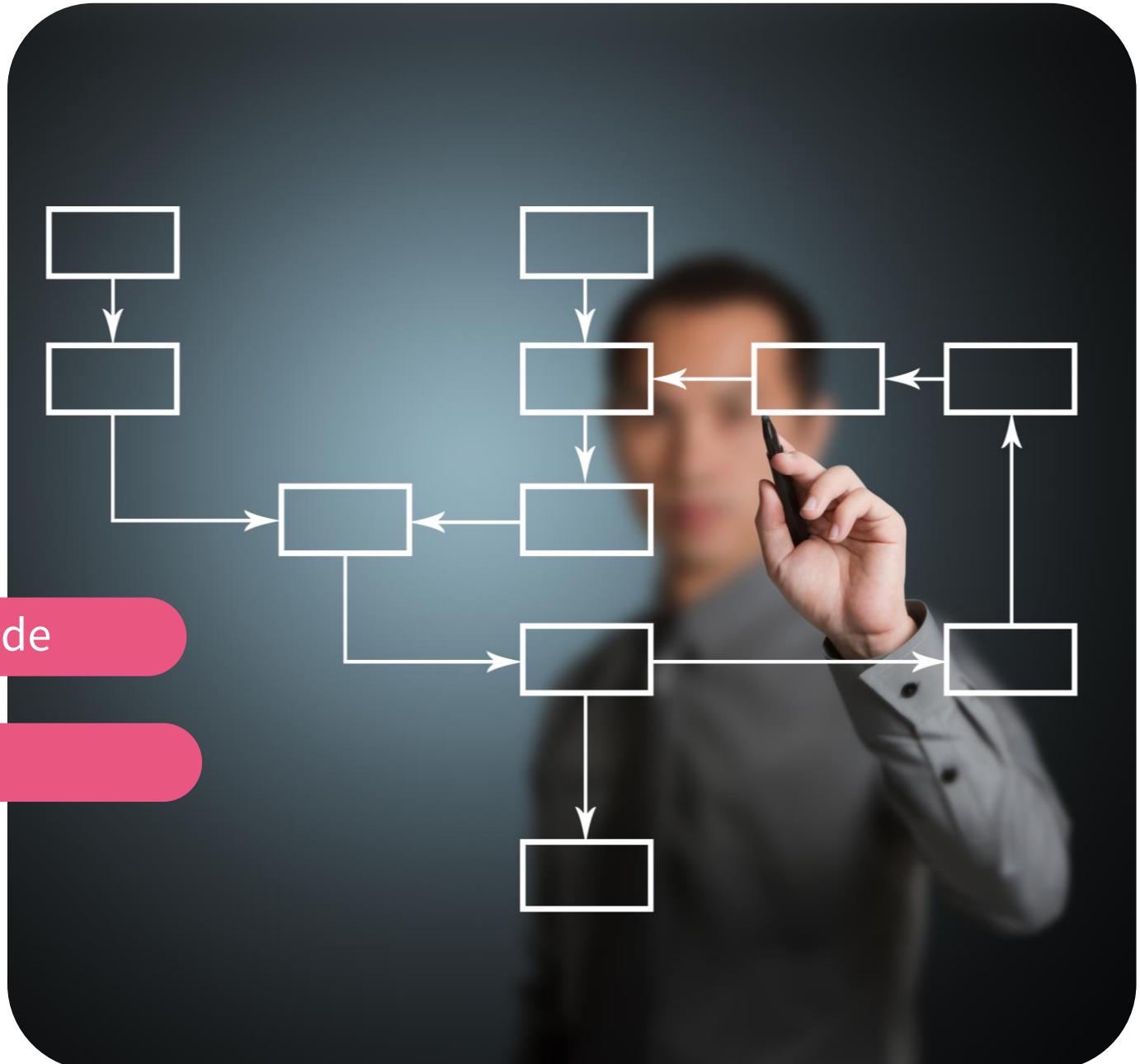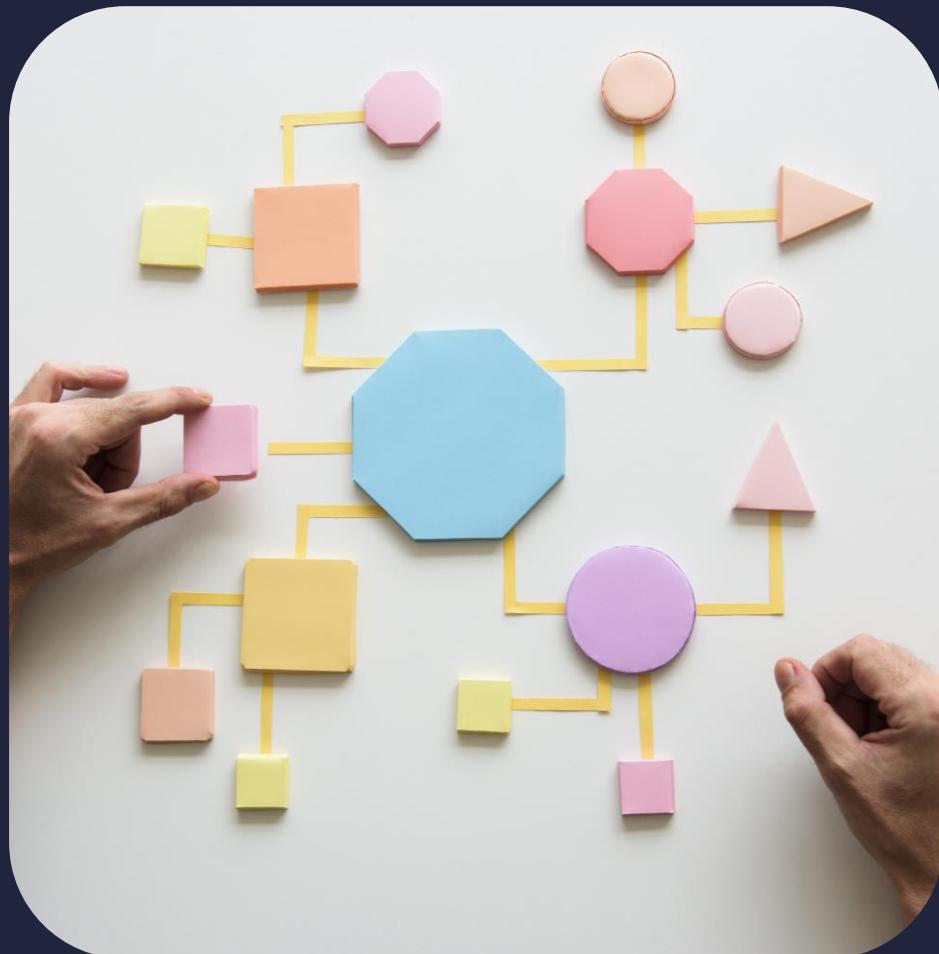
Flow charts

Often used in conjunction with pseudocode

Flowcharts help to include all the steps
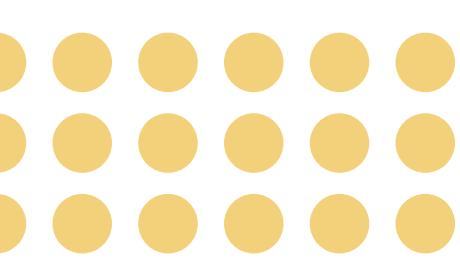
Flow charts

- Start and stop points of a sequence are denoted by a rounded rectangle

- Process is denoted by a rectangle

- Decision, yes or no, is denoted by a rhombus

- Input or output denoted by a parallelogram

- Use a circle to jump from one point to another

- Use arrows for other connections

Name	Symbol	Usage
Start/stop or end		The beginning and end of the sequence/program
Process		An instruction
Decision		A decision typically an *if* statement
Input/output		Data being received by a computer or data produced by the computer
Connector		A jump from one point in the sequence to another
Direction of flow		Direction of the flow of the program
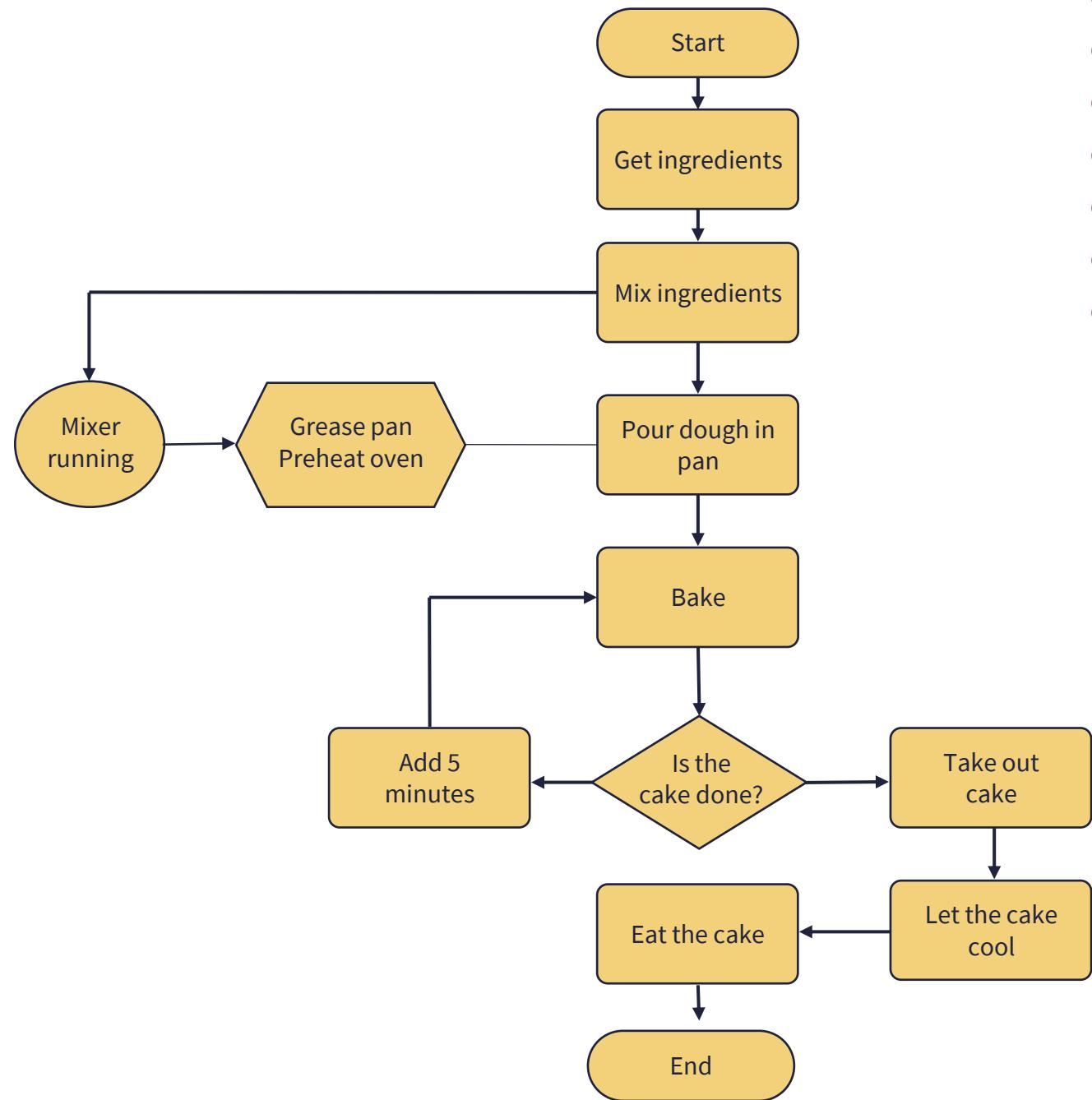
Baking a cake

Program flowchart

Write pseudocode for a program that asks for 10 numbers from the user, then look at the numbers and displays only even numbers.
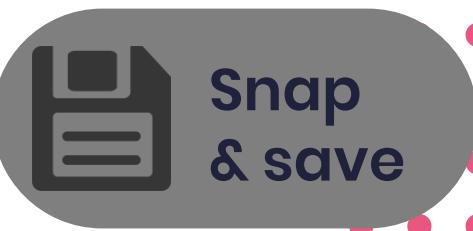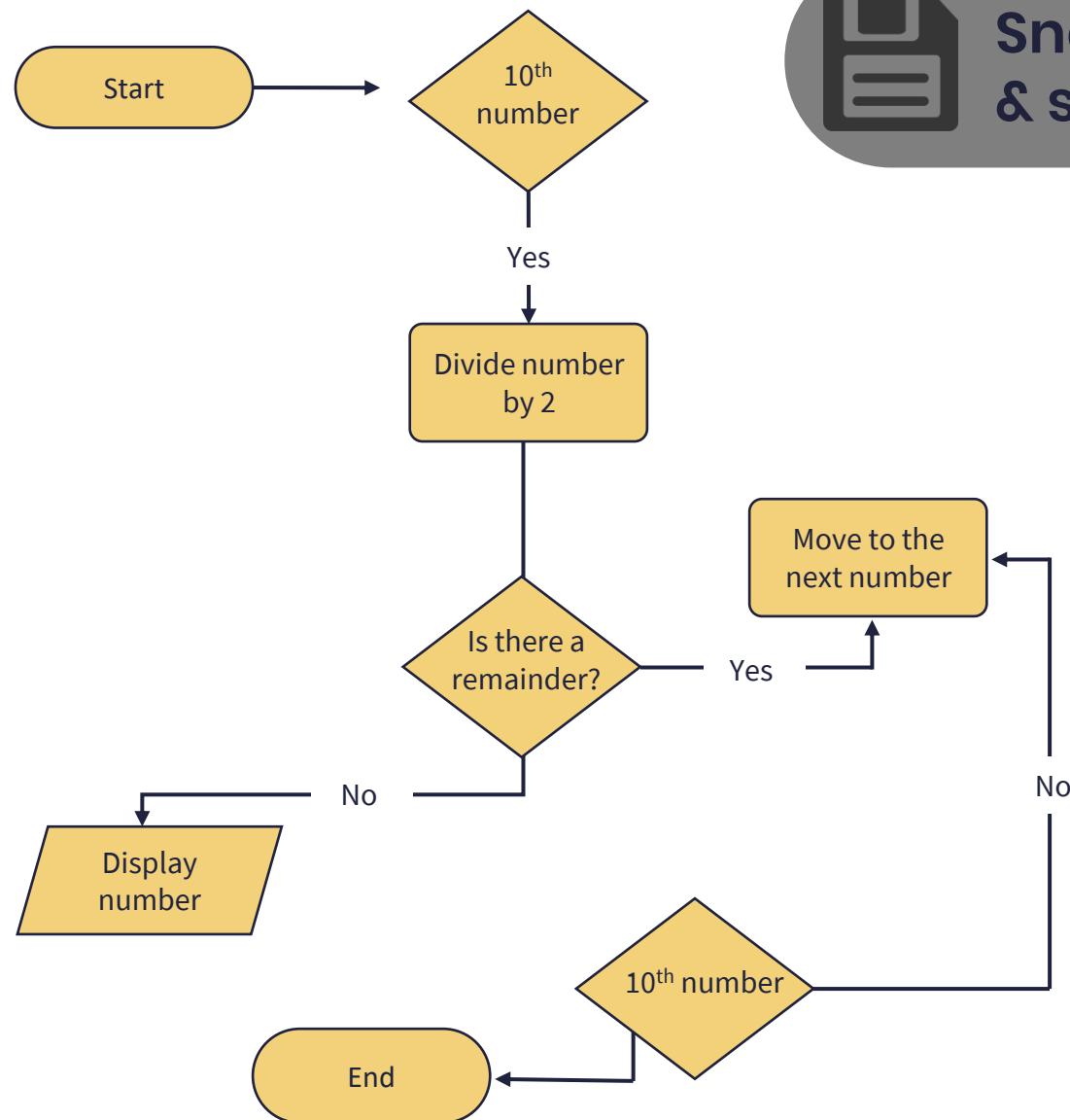
Pseudocode example

Start
PROCEDURE read numbers
       WHILE recorded numbers <10
       DO:
Ask user for number
Store number
PROCEDURE find even numbers
       WHILE recorded numbers <10
       DO:
Fetch number
Divide number by 2
       IF number has remainder
       THEN ignore number
       ELSE display number

END

Pseudocode example

Start
PROCEDURE read numbers
        WHILE recorded numbers <10
        DO:
Ask user for number
Store number
PROCEDURE find even numbers
        WHILE recorded numbers <10
        DO:
Fetch number
Divide number by 2
        IF number has remainder
        THEN ignore number
        ELSE display number
END

Keywords in pseudocode

INPUT: used for data obtained from the user

READ/GET: used when reading data from a data file
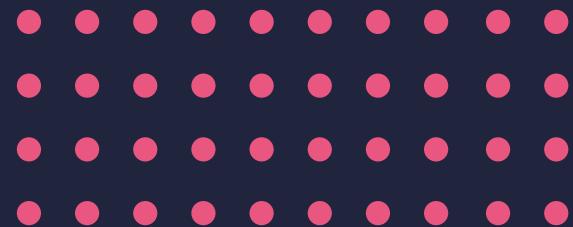
PRINT/DISPLAY/SHOW: shows your output

COMPUTE/CALCULATE/DETERMINE: used to calculate the result of an expression

SET/INIT: used to initialise values

INCREMENT/BUMP: used when increasing the value of a variable

DECREMENT: used when reducing the value of a variable
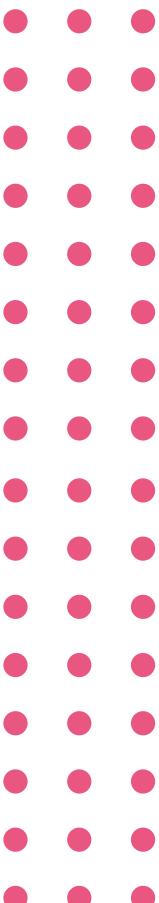
Creating pseudocode & flowcharts

Task

- Write pseudocode for a programme that asks the user for the description of a shape.

- Based on certain keywords, it will identify the shape, then ask the user for dimensions.

- The program then proceeds to calculate the surface area and/or volume of the identified shape.

- It must then display the name of the shape as well as the results of the calculation to the user.

Establish what exactly needs to be done

Start with the requirements

Need a reference to gauge whether you have solved the given problem

Requirements for shape calculation problem

- Ask the user for the description of a shape.

- Ask the user for the dimensions of the shape.

- The program needs to know which dimensions are applicable to the specified shape.

- Specify which shapes the user can input for calculation.

- Can reject the shapes that it is not programmed to calculate.

- Tell whether a shape is a flat shape or a solid shape.
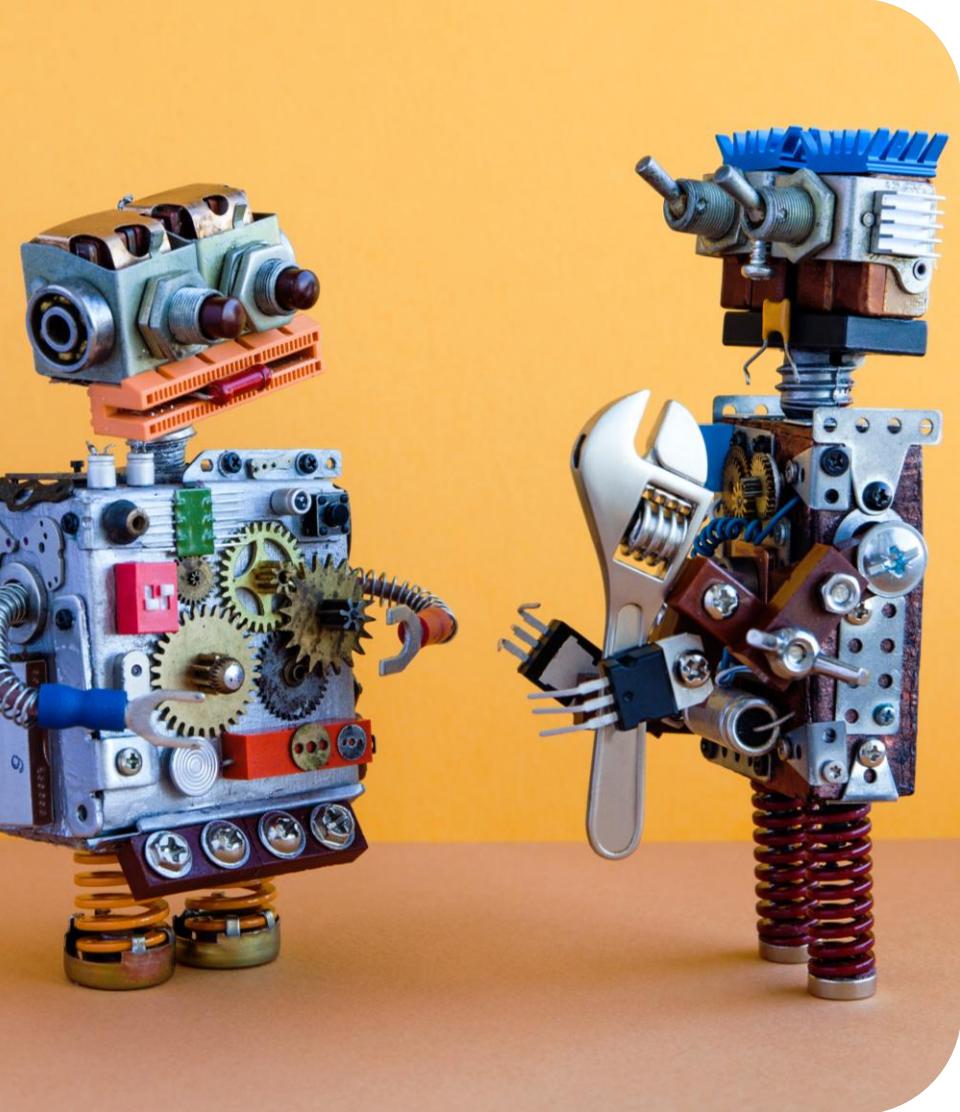
Requirements

- Calculate the area of flat shape or volume of solid shape

- Display the name of the shape, volume and surface area

Provisional pseudocode

PROCEDURE calculate
surface_area_and_volume
START
READ shape name from user
STORE shape_name
IF shape_name is circle THEN
READ radius
Area=pi*radius*radius
DISPLAY shape_name
DISPLAY area
ENDIF
ELSE IF shape_name is sphere THEN
Read radius
Area=4*pi*radius*radius
Volume=4/3*pi*radius*radius*radius
DISPLAY shape_name
DISPLAY area
DISPLAY volume
ENDIF
ELSE IF shape_name is square THEN
READ side
Area=side*side
DISPLAY shape_name
DISPLAY area
ENDIF

ELSE IF shape_name is cube THEN
READ side
Area=side*side
Volume=side*side*side
DISPLAY shape_name
Display area
ENDIF
ELSE IF shape_name is rectangle THEN
READ side1
READ side2
Area=side1*side2
DISPLAY shape_name
DISPLAY area
ENDIF
ELSE IF shape_name is cuboid THEN
READ side1
READ side2
READ height
Area=side1*side2
Volume=side1*side2*height
DISPLAY shape_name
DISPLAY area
DISPLAY height
ENDIF
END PROCEDURE

Refining pseudocode

```
PROCEDURE calculate
        surface_area_and_volume
START
READ shape name from user
STORE shape_name
IF shape_name is circle THEN
            READ radius
            Area=pi*radius*radius
            DISPLAY shape_name
            DISPLAY area
ENDIF
ELSE IF shape_name is sphere THEN
        Read radius
        Area=4*pi*radius*radius
        Volume=4/3*pi*radius*radius*radius
        DISPLAY shape_name
        DISPLAY area
        DISPLAY volume
ENDIF
ELSE IF shape_name is square THEN
        READ side
        Area=side*side
        DISPLAY shape_name
        DISPLAY area
ENDIF

ELSE IF shape_name is cube THEN
        READ side
        Area=side*side
        Volume=side*side*side
        DISPLAY shape_name
        Display area
ENDIF
ELSE IF shape_name is rectangle THEN
        READ side1
        READ side2
        Area=side1*side2
        DISPLAY shape_name
        DISPLAY area
ENDIF
ELSE IF shape_name is cuboid THEN
        READ side1
        READ side2
        READ height
        Area=side1*side2
        Volume=side1*side2*height
        DISPLAY shape_name
        DISPLAY area
        DISPLAY volume
ENDIF
END PROCEDURE
```

Refining pseudocode

- DISPLAY shape_name
- DISPLAY area
- DISPLAY volume

```
PROCEDURE calculate
surface_area_and_volume
START
READ shape name from user
STORE shape_name
IF shape_name is circle THEN
READ radius
Area=pi*radius*radius
DISPLAY shape_name
DISPLAY area
ENDIF
ELSE IF shape_name is sphere THEN
Read radius
Area=4*pi*radius*radius
Volume=4/3*pi*radius*radius*radius
DISPLAY shape_name
DISPLAY area
DISPLAY volume
ENDIF
ELSE IF shape_name is square THEN
READ side
Area=side*side
DISPLAY shape_name
DISPLAY area
ENDIF
```

```
ELSE IF shape_name is cube THEN
READ side
Area=side*side
Volume=side*side*side
DISPLAY shape_name
Display area
ENDIF
ELSE IF shape_name is rectangle THEN
READ side1
READ side2
Area=side1*side2
DISPLAY shape_name
DISPLAY area
ENDIF
ELSE IF shape_name is cuboid THEN
READ side1
READ side2
READ height
Area=side1*side2
Volume=side1*side2*height
DISPLAY shape_name
DISPLAY area
DISPLAY volume
ENDIF
END PROCEDURE
```

Refining pseudocode

- Create a separate routine.

```
PROCEDURE calculate
    surface_area_and_volume
START
READ shape name from user
STORE shape_name
IF shape_name is circle THEN
        READ radius
        Area=pi*radius*radius
ENDIF
ELSE IF shape_name is sphere THEN
    Read radius
    Area=4*pi*radius*radius
    Volume=4/3*pi*radius*radius*radius
ENDIF
ELSE IF shape_name is square THEN
    READ side
    Area=side*side
ENDIF
ELSE IF shape_name is cube THEN
    READ side
    Area=side*side
    Volume=side*side*side
ENDIF
```

```
ELSE IF shape_name is rectangle THEN
    READ side1
    READ side2
    Area=side1*side2
ELSE IF shape_name is cuboid THEN
    READ side1
    READ side2
    READ height
    Area=side1*side2
    Volume=side1*side2*height
ENDIF
ROUTINE display_results
DISPLAY shape_name
        DISPLAY area
        If volume>0 THEN
        DISPLAY volume
END PROCEDURE
```

```mermaid
flowchart

START --> Read_shape[Read shape]
Read_shape --> Store_shape[Store shape name]

Store_shape --> Circle{Circle?}
Circle -- Yes --> CircleProc[Circle Procedure]
Circle -- NO --> Sphere{Sphere?}
Sphere -- Yes --> SphereProc[Sphere procedure]
Sphere -- NO --> Square{Square?}
Square -- Yes --> SquareProc[Square procedure]
Square -- NO --> Cube{Cube?}
Cube -- Yes --> CubeProc[Cube procedure]
Cube -- NO --> Rectangle{Rectangle?}
Rectangle -- Yes --> RectangleProc[Rectangle procedure]
Rectangle -- NO --> Cuboid{Cuboid?}
Cuboid -- Yes --> CuboidProc[Cuboid procedure]
Cuboid -- NO --> ShapeNot[Shape not defined]

CircleProc --> Display[Display results]
SphereProc --> Display
SquareProc --> Display
CubeProc --> Display
RectangleProc --> Display
CuboidProc --> Display

Display --> Volume{Is volume>0?}
Volume -- Yes --> DisplayVol[Display volume]
Volume -- NO --> END
DisplayVol --> END
ShapeNot --> END
```

Benefits of using a flowchart AND pseudocode

Flowchart clearly sets out the steps to follow to solve the problem

Pseudocode details exactly how procedures are carried out

Challenge »

You work for a software company and they have asked you to create a module for their new program.

The program should count the number of words in a document, then display the result to the user.

Challenge »

You are a lecturer at an educational institution.

Design a system that records each student's surname and average mark.

The system should be able to record average marks for 30 students and display these marks in a list.

The system should then calculate the class average mark and display this below the class list.

Write pseudocode and create a flowchart for this system.