AS Computer Studies

Algorithm Design 1

Starter - Lateral thinking

- Mel Colly stared through the dirty soot-smeared window on the 26th floor of the office tower.

An eagle, an elephant, and a walleye each have two.

A tiger, a moose, a bear, a turtle, and a snake have one.

- Overcome with depression he slid the window open and jumped through it.

Humans, birds, cats, and dogs don't have any.

- Miraculously after he landed he was completely unhurt.

What is this thing?

- Since there was nothing to cushion his fall or slow his descent, how could he have survived?

The letter "e"

Window washing, he opened the window and jumped inside.

Starter 2

- What are the processes needed to write a program for calculating your pay for a week's work? Write down ALL steps!

- 5 minutes we will review…
  - Give your algorithm to someone else. They have to then follow your algorithm EXACTLY, for calculating their weekly pay. They cannot make any assumptions!
  - They must only do what your instructions say – nothing more.

Starter Review

- Did you include every little step?
- Did you make assumptions about the other person's knowledge?
- Did you refine it (remember Stepwise Refinement!)
- When writing a program to be run on computers they require **EVERY** bit of detail. You cannot make any assumptions. Everything must be accounted for.

Objectives

- Understand what an **algorithm** is.
- Recap basic algorithm knowledge.
- Understand the process of writing algorithms in **structured English**.
- Understand elements of computing: **Sequence**, **Selection** and **Iteration**

Algorithm

- What is an algorithm? Why do we bother using them?
- What is a program?
- An **algorithm** is a description independent of any programming language, of a process that achieves some task. It is a step-by-step procedure for solving a problem. (it has no programming language in it)
- They allow us to **clearly understand a problem**, using stepwise refinement to define all the elements of a problem and ensure they are solved efficiently.
- A program is a description in a programming language of a process that achieves some useful result (in other words, it **takes the algorithm and turns it into a program**).

An example work-through

- You have examples of Algorithm Design, which you completed in previous lessons.
- Let's look at this algorithm design for washing a car, using stepwise refinement.
- EXAMPLE

Why bother

- Ideas? Give me an example. Think more complex projects than washing a car!
- Any computer program which does a worthwhile job in the real world is a complex piece of work.
- In order for it to function correctly, a mass of detailed instructions has to be dealt with and organised into the correct sequence for the computer to execute.
- Imagine getting a few tools, some bits of metal, assorted screws, some electrical bits and pieces, and assemble a Jumbo Jet by trying things until it flies. **WOULDN'T HAPPEN**
- Imagine a skilled carpenter might put together a box without much thought, just based on experience - but you wouldn't want even the most experienced carpenter putting in a fitted kitchen for you unless there were detailed plans and drawings from which to work.
- Thousands of people think that it is possible to produce a program just by sitting down at a terminal and hacking away until it "works". This is **utter nonsense**.

The Different Parts

- **Sequence** (flow from one line of code to the next in order) (e.g. step by step).
- **Selection** (flow to a line selected by some condition in the code) If…Then…
- **Iteration** (repetition of a block of statements as long as some condition is true)
- **Assignment –** give / store a value to a declared variable.
- **Let's look back at the car algorithm to show each of these.**

Algorithms - Structured English

- Based around verbs which clearly define what a program will do.

- Examples words (are any of these unclear?)
  - **MULTIPLY**
  - **SET**
  - **DO**
  - **ADD**
  - **SUBTRACT**
  - **UPDATE**
  - **WRITE**
  - **OUTPUT**
  - **PRINT**

Please write these down

Structured English SEQUENCE Example

GET score1, score2, score3

SET totalScore = score1 + score2 + score3

SET averageScore = totalScore / 3

UPDATE student-record from file

WRITE updated student-record to Student file

OUTPUT totalScore, averageScore

PRINT student-record

Draw the structure diagram (hierarchy chart) for calculating and printing the total & average scores for 3 tests.

Structured English Example

Imagine if you had lots of different things you could do to student information. (e.g. Delete, Edit, Print, Update Score etc). Each would be in it's own **MODULE**.

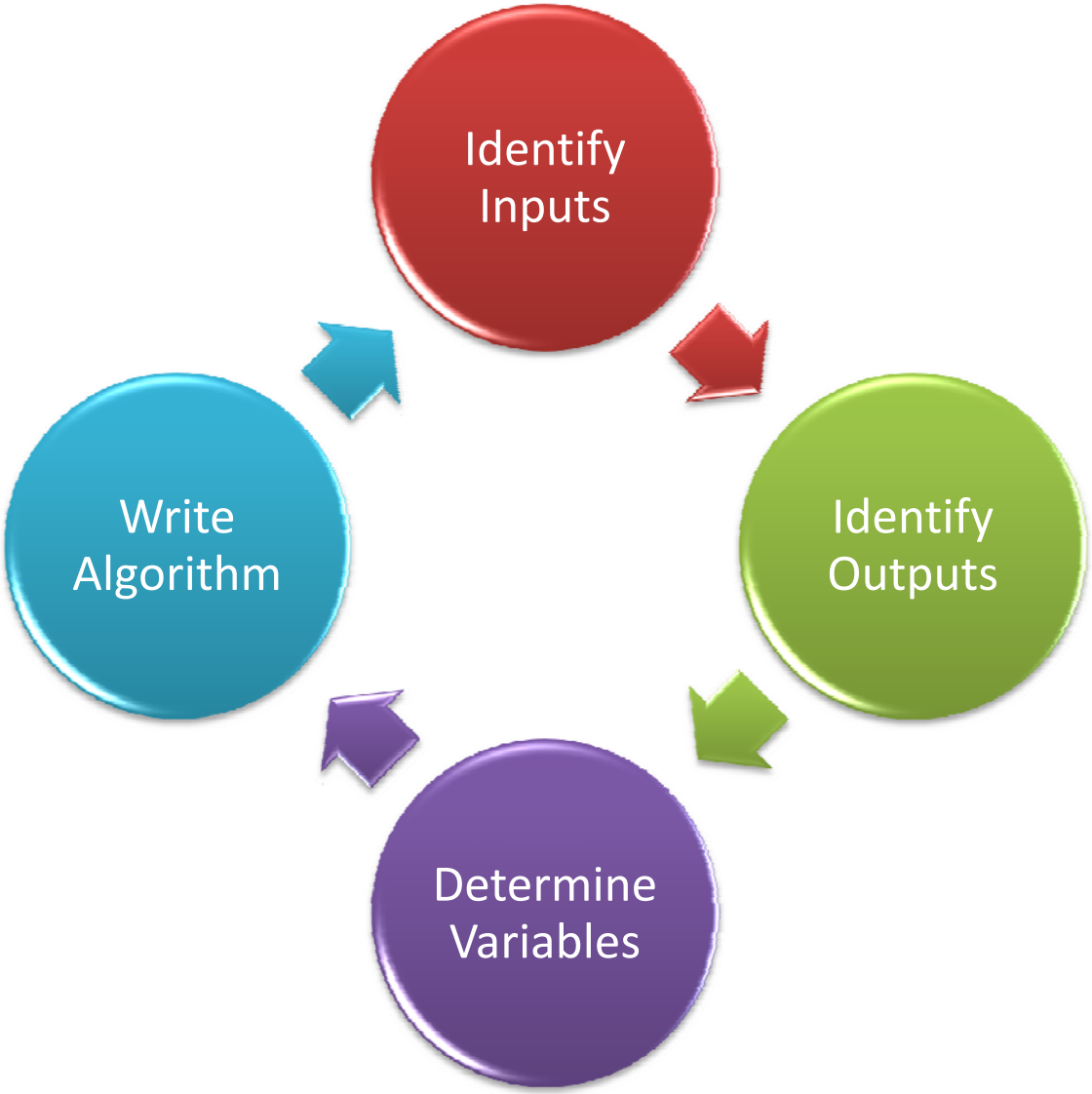
A module just gives a name to a **block of code which performs a specific task**.

If we named the code before **CalculateScore**, to execute those commands, all you have to say now is:

DO CalculateScore

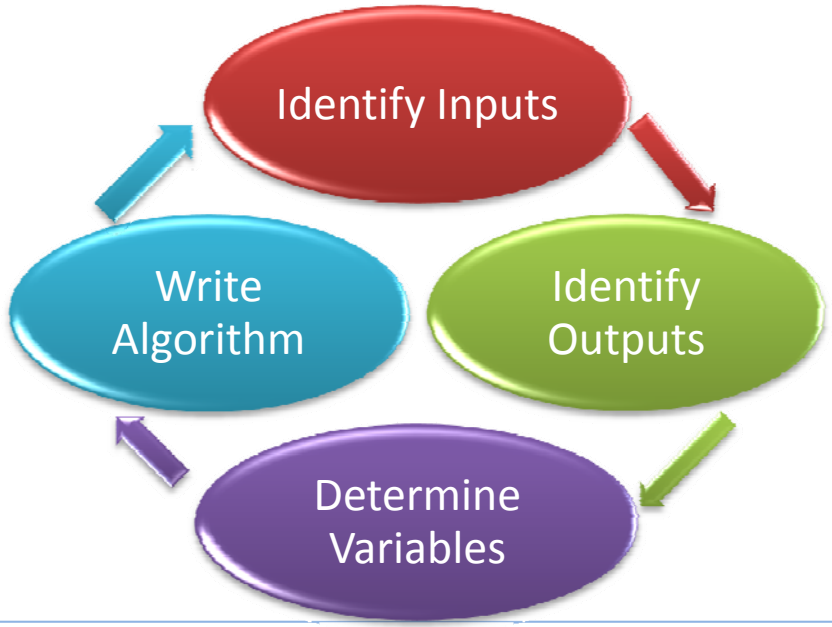
CalculateScore
GET score1, score2, score3
UPDATE student-record from file
SET totalScore = score1 + score2 + score3
SET averageScore = totalScore / 3
WRITE updated student-record to Student file
OUTPUT totalScore, averageScore
PRINT student-record

Algorithms for Computer Based Problem Solving



Worked Example 1

- Super Star Games Design would like a system whereby:
  - the price of a game is entered and
  - the VAT (15.5%) is calculated.
  - The system should display the VAT payable,
  - plus the price of the game and
  - the price of the game inclusive of VAT.
- Write an algorithm which shows this, in Pseudocode.



SET

GET

MULTIPLY

ADD

SUBTRACT

INPUT

OUTPUT

PRINT

Solution

- Super Star Games Design would like a system whereby the price of a game is entered and the VAT (15.5%) is calculated. The system should display the VAT payable, plus the price of the game and the price of the game inclusive of VAT.

- DO TaxPriceCalculator

- TaxPriceCalculator

  INPUT gamePrice

  SET VAT to 15.5%

  MULTIPLY gamePrice by VAT to get gameVAT

  ADD gamePrice to gameVAT to get fullGamePrice

  OUTPUT gamePrice, gameVAT, fullGamePrice

Structured English: Sequence Tasks
Group Work (10 mins, choose ONE scenario)

1. Bradleys Bank would like a system which calculates loan repayment amounts. A loan value would be entered, as would an interest rate (e.g. 6%). The interest would be added to the loan and a monthly payment would be calculated. The system should display and print the loan amount, the amount repayable and the monthly repayments.

2. Outline UK would like an online pay as you go calculator. For every £1 which is topped up, they will give customers 20 texts and 20 minutes. A calculator is needed to display the total text and minutes for a given value of top up credit.

3. A school would like a monthly invoice of its printer usage. When given information, an invoice would be produced to calculate the total amount owed for mono printing and colour printing, along with a combined total. Mono = 6p per sheet. Colour = 8p per sheet. The total printed for each printer would be provided.

Objectives Review

- Understand what an **algorithm** is.
- Recap basic algorithm knowledge.
- Understand the process of writing algorithms in **structured English**.
- Understand elements of computing: **Sequence**, **Selection** and **Iteration**

Plenary

- Teach back.
- You Say We Play
  - Class describes the phrase with an EXAMPLE to the player.
  - The player has to say what word they think is on the board.

Acknowledgements

- Information kindly provided by the University of Teesside, Carole Wagstaff – An Introduction to Visual Programming.