Lecture 2

Algorithm, Process Flow & Pseudocode

Dr M Kasim A Jalil

Faculty of Mechanical Engineering

UTM

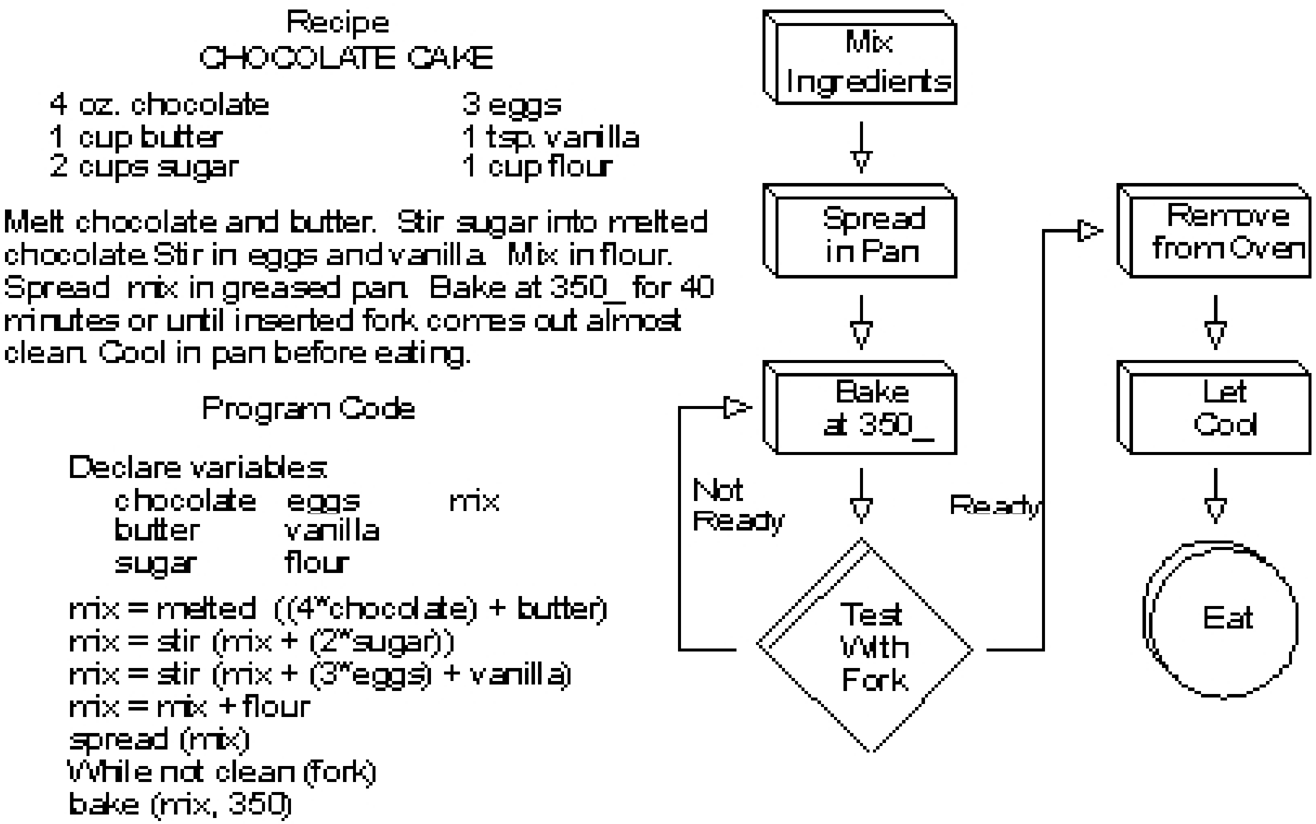
Algorithm

- A formula or set of steps for solving a particular problem. To be an algorithm, a set of rules must be unambiguous and have a clear stopping point. Algorithms can be expressed in any language, from natural languages like English or French to programming languages like C.
- We use algorithms every day. For example, a recipe for baking a cake is an algorithm. Most programs, with the exception of some artificial intelligence applications, consist of algorithms. Inventing elegant algorithms -- algorithms that are simple and require the fewest steps possible -- is one of the principal challenges in programming.

Assignment

- Revisit the Chicken Curry problem
- Think of a better way to represent the process in a systematic flow chart

An example of Algorithm



Recipe
CHOCOLATE CAKE

4 oz. chocolate	3 eggs
1 cup butter	1 tsp. vanilla
2 cups sugar	1 cup flour

Melt chocolate and butter. Stir sugar into melted chocolate. Stir in eggs and vanilla. Mix in flour. Spread mix in greased pan. Bake at 350_ for 40 minutes or until inserted fork comes out almost clean. Cool in pan before eating.

Program Code

```
Declare variables:
    chocolate   eggs      mix
    butter      vanilla
    sugar       flour
mix = melted ((4*chocolate) + butter)
mix = stir (mix + (2*sugar))
mix = stir (mix + (3*eggs) + vanilla)
mix = mix + flour
spread (mix)
While not clean (fork)
bake (mix, 350)
```

Pseudocode

- An outline of a program, written in a form that can easily be converted into real programming statements. For example, the pseudocode for a bubble sort routine might be written:

```
while not at end of list
compare adjacent elements
if second is greater than first
switch them
get next two elements
if elements were switched
repeat for entire list
```

- Pseudocode cannot be compiled nor executed, and there are no real formatting or syntax rules. It is simply one step - an important one - in producing the final code. The benefit of pseudocode is that it enables the programmer to concentrate on the algorithms without worrying about all the syntactic details of a particular programming language. In fact, you can write pseudocode without even knowing what programming language you will use for the final implementation.

Computer Coding

- A program is a sequence of instructions to the computer for it to solve a particular problem. A set of programs is called code.
- Programs are written in some programming language – e.g. Fortran, Basic, C++, Pascal, Java, Matlab.
- Programs are stored in files – which are a sequence of bytes which is given a name and stored on a disk.

Computer Coding (cont'd)

- A bit or a binary digit is the smallest unit of data storage. It has the value of either 0 or 1 only.
- A byte is a sequence of 8 bits, representing 256 different values. E.g. of a byte is:

  0 1 0 1 1 1 0 1

  A byte could represent a character like a letter or a number or a symbol.
- A word is a larger group of bytes, maybe two bytes (16 bits).

Computer Coding (cont’d)

- A program is a file containing a sequence of “statements”, each of which tells the computer to do a specific action.
- Once a program is run or executed the commands are followed and actions occur in a sequential manner.
- If the program is designed to interact with the outside world, then it must have input and output.

Computer Coding (cont'd)

- A program is said to have a bug if it contains a mistake or it does not function in the way it is intended to.
- Bugs can happen both in the logic of the program, as well as in the commands.
- In order that the program perform the exact actions it is intended to do, before the actual program is written an algorithm for solving the problem must first be outlined.

Computer Coding (cont'd)

- An algorithm is a general sequence of the logical steps in solving a specific problem.
- A flowchart is a graphical representation of the algorithm.
- A pseudocode gives a more detailed, step-by-step set of instructions required to solve the problem. It is normally written in everyday language.

Good Algorithms

- Each step must be deterministic – it cannot be ambiguous or leaving things to chance.
- The process must end after a finite number of steps – it must not be open-ended.
- The algorithm must be general enough to deal with any contingency.

Example Problem #1

- Given temperature in degrees Fahrenheit, the temperature in degrees Kelvin is to be computed and shown.
- Formula:

$$\boxed{T_K = \left(\frac{T_F - 32}{1.8}\right) + 273.15}$$

- TK and TF are temperatures in degrees Kelvin and Fahrenheit respectively

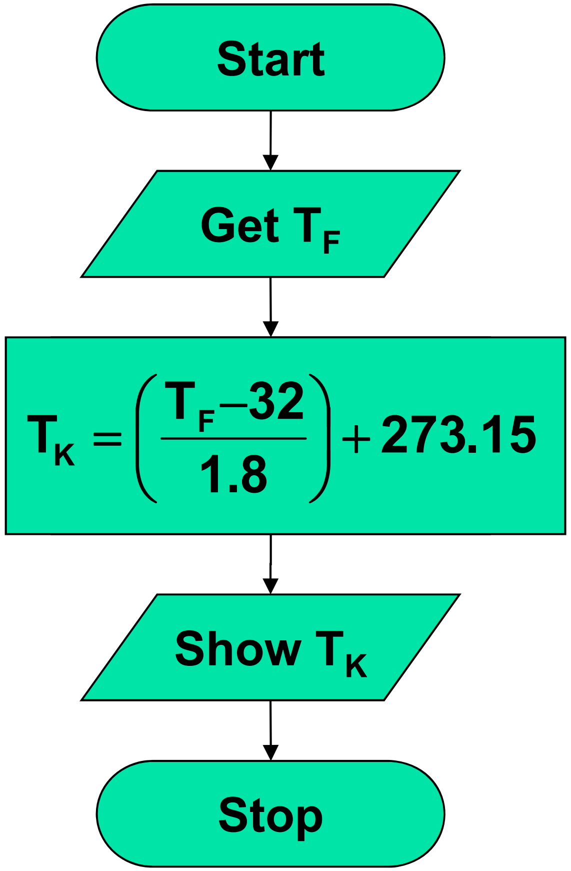
Algorithm

- Start
- Get the temperature in Fahrenheit
- Compute the temperature in Kelvin using the formula

$$\boxed{T_K = \left(\frac{T_F - 32}{1.8}\right) + 273.15}$$

- Show the temperature in Kelvin
- Stop

Flowchart



Pseudocode

```
1. Start
2. Get TF
3. $TK = (TF-32)/1.8 + 273.15$
4. Show TK
5. Stop
```

Example Problem #2

- Given a set of numbers, calculate their sum and the average value (mean).
- Formula:

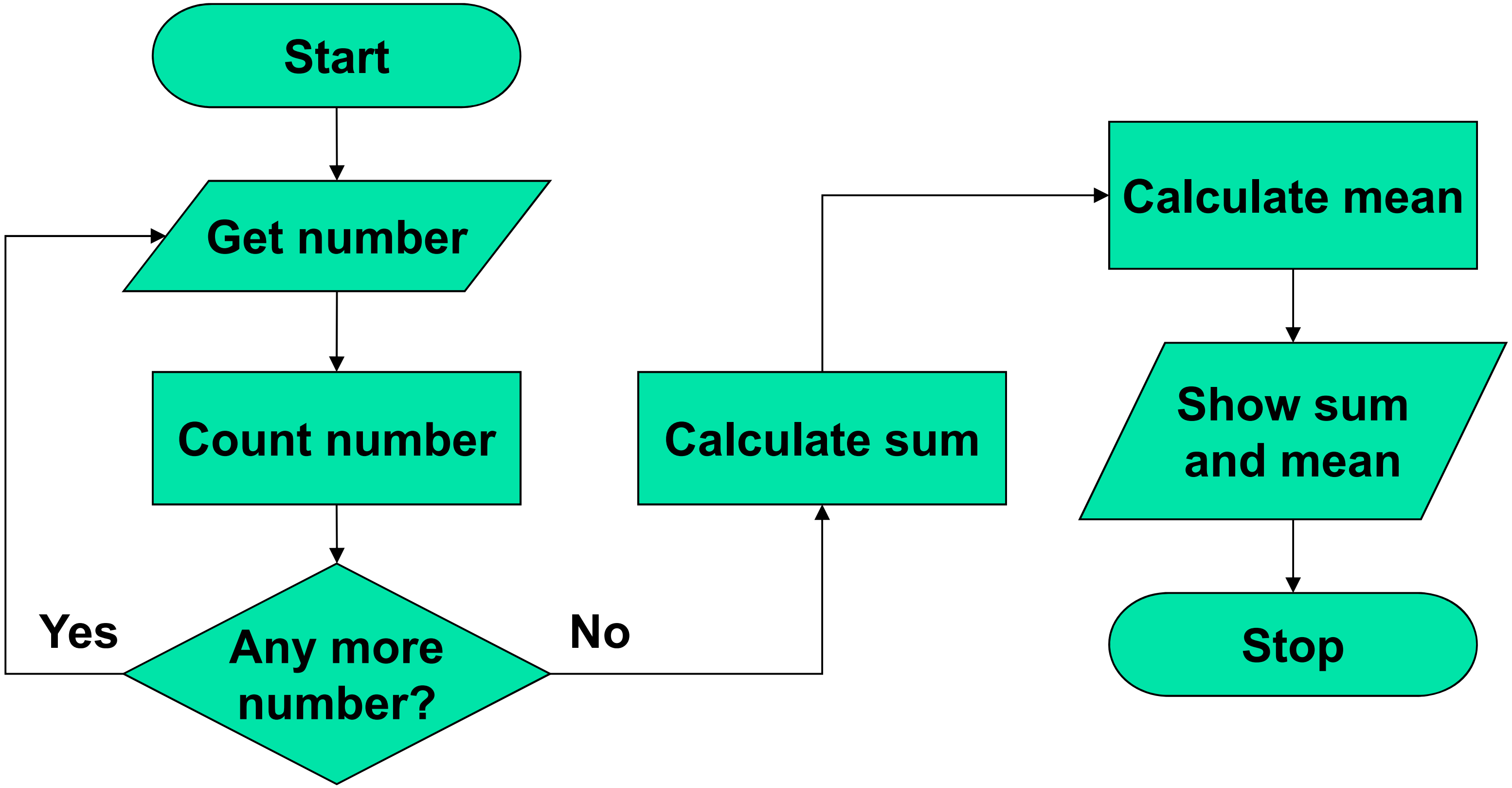
$$\boxed{\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i}$$

- n is the number of numbers in the set

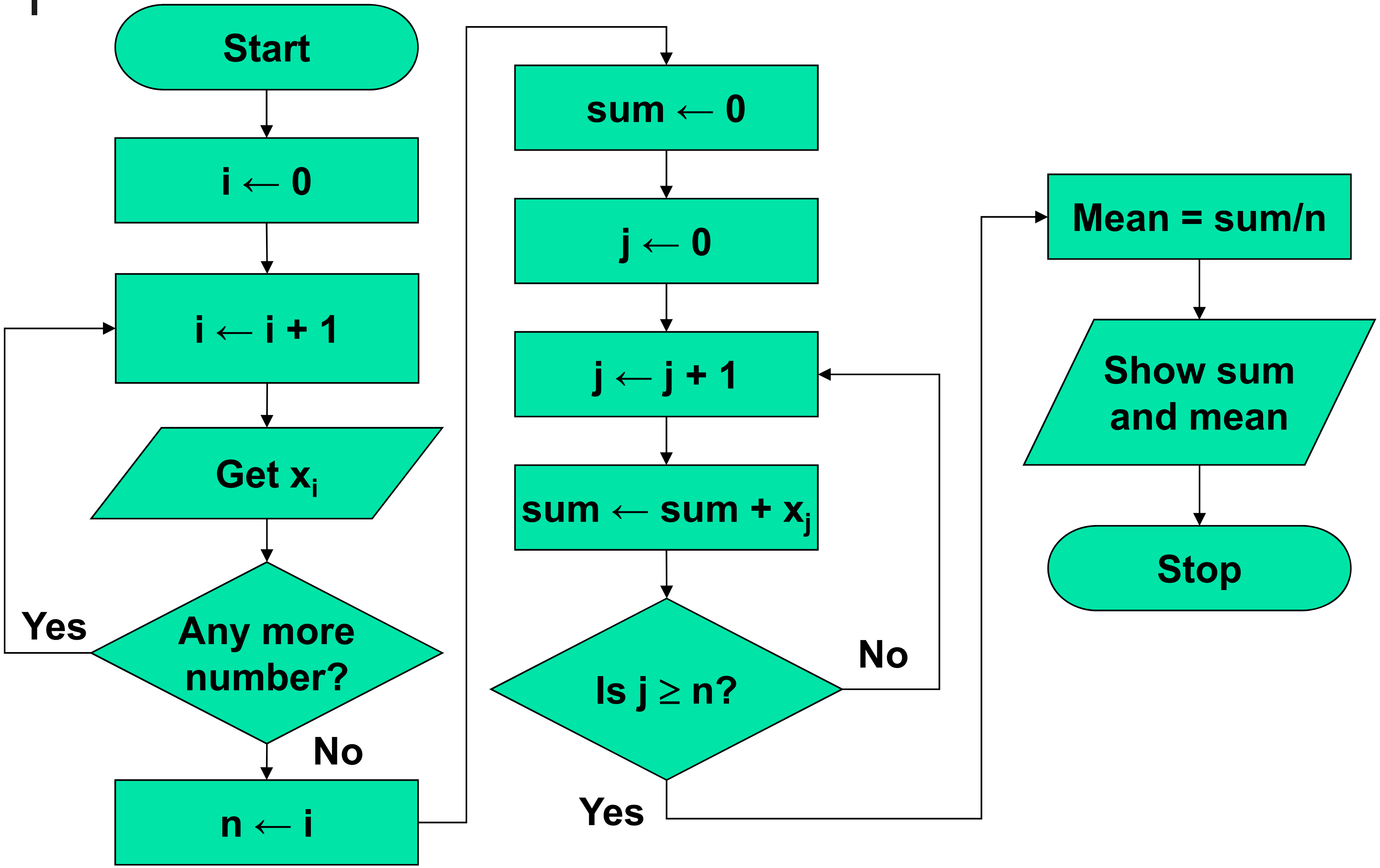
Algorithm

1. Start
2. Get one number in the set
3. Count the numbers as it is obtained
4. If there are still numbers to be obtained, go back to step 2.
5. Sum the numbers in the set
6. Divide the sum by the number of numbers in the set to get the average
7. Show the sum and the average
8. Stop

Flowchart


Start
Get number
Count number
Any more number?
Yes
No
Calculate sum
Calculate mean
Show sum and mean
Stop

Detailed Flowchart

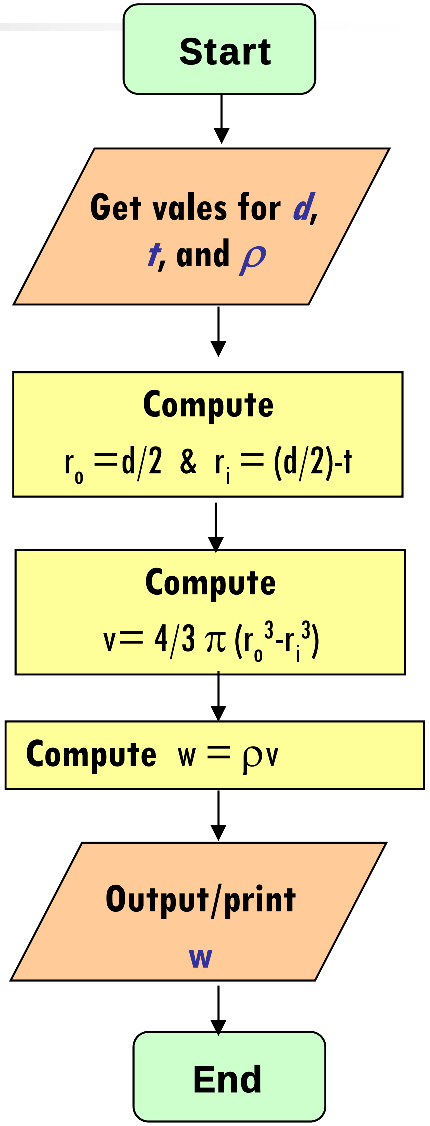


Example 3

Define the steps to compute the weight *w* of a hollow sphere of diameter *d*, wall thickness *t*, and density *ρ*, using the following equations:

- $r_o = d/2$
- $r_i = (d/2)-t$
- $v = 4/3\,\pi\,(r_o^3 - r_i^3)$
- $w = \rho v$

1. Get vales for *d*, *t*, and *ρ*
2. Compute the inner and outer radii
   - $r_o = d/2$
   - $r_i = (d/2)-t$
3. Compute the volume of sphere
   - $v = 4/3\,\pi\,(r_o^3 - r_i^3)$
4. Compute the weight of sphere
   - $w = \rho v$
5. Ouput/print w



Start

Get vales for *d*, *t*, and *ρ*

Compute $r_o = d/2$ & $r_i = (d/2)-t$

Compute $v = 4/3\,\pi\,(r_o^3 - r_i^3)$

Compute $w = \rho v$

Output/print w

End

Example 4

Name	HW	Test1	Test2	Project
Popal	80	56	75	95
Ah Beng	95	34	99	87
Kevin	90	57	88	79

- **Evaluation Scheme:**
  - HW = 30%
  - Test 1 & Test 2 = 25% each
  - Project = 20%
- **Calculate the individual mark**
- **Show the step-by-step process**
- **Construct the flowchart**

Example 5

Name	Marks
John	86
Kalvinder	76
Amit	72
Aaron	89
Mehmet	95
Atilla	91

- Grading Scheme
  - Above 95 = A
  - 90-94 = A-
  - 85-89 = B+
  - 80-84 = B
  - 75-89 = B-
  - 70-74 = C+
  - 65-69 = C
  - 60-64 = C-
- Specify the grade for each student.
- Show your step-by-step process
- Draw your flowchart