AL - Mustansiriyah University, Collage of Education
Computer Science Department

1st Class
2018-2019
Structured Programming
البرمجة المهيكلة

مدرس المادة:أ.م.د. سوسن عبد الهادي محمود(للدراسةالمسائي)
أ.م. امل عباس كاظم (للدراسة الصباحي)

Lecture 1: Introduction

What is a Computer?

Computer
A device capable of performing computations and making logical decisions in a very fast manner.

Computer programs
Sets of instructions that control a computer's processing of data

Hardware
Various devices comprising a computer
Examples: keyboard, screen, mouse, disks, memory, CD-ROM, and processing units

Software
Programs that run a computer

Computer Organization

Input unit
Obtains information from input devices (keyboard, mouse)

Output unit
Outputs information (to screen, to printer, to control other devices)

Memory unit
Rapid access, low capacity, stores input information

Arithmetic and logic unit (ALU)
Performs arithmetic calculations and logic decisions

Central processing unit (CPU)
Supervises and coordinates the other sections of the computer

Secondary storage unit
Cheap, long-term, high-capacity storage, stores inactive programs

Programming

A digital computer is a useful tool for solving a great variety of problems. A solution to a problem is called an algorithm; it describes the sequence of steps to be performed for the problem to be solved.

An **algorithm** is expressed in abstract terms. To be intelligible to a computer, it needs to be expressed in a language understood by it. The only language really understood by a computer is its own machine language.

Programs expressed in the machine language are said to be executable. A program written in any other language needs to be first translated to the machine language before it can be executed.

1. Programming Languages (Machine languages)

- Strings of numbers giving machine specific instructions.
- Computers can only understand this language.

Example:

10100011001001
11111111111111
00000001110100

- Machine dependent: every machine has its own language.
- Hard to be understood by humans.
- Hard to be used in programming.
- Too slow and tedious.

2. Programming Languages (Assembly languages)

- English-like abbreviations representing elementary computer operations so it is easier to be understood by humans.
- Translated or converted into machine language via assemblers.
- Also, it is slow and hard to be used in programming.
- Machine dependent.

Example:

```
LOAD    BASEPAY
ADD     OVERPAY
STORE   GROSSPAY
```

3. High-level languages

- Similar to everyday English, use mathematical notations.
- Translated into machine language via compilers (compile the whole program at once)
- Interpreters are used to execute high level languages without need to compile them into machine language and it execute single line at a time.
- Compiled programs are faster than the interpreted ones.
- Fast and easy for programming.
- Machine independent.

Example: Valu1= A+ B

Example of High-level Languages

- C and C++.
- Java
- Visual basic 6/.Net
- C#.Net
- FORTRAN
- COBOL
- Pascal

An algorithm is a procedure for solving problems. specifies a series of steps that perform a particular computation or task. In order to solve a mathematical or computer problem. An algorithm includes calculations, reasoning and data processing. Algorithms can be presented by natural languages, pseudo code and flowcharts, etc.

Algorithms were originally born as part of mathematics – the word “algorithm” comes from the Arabic writer "Muḥammad ibn Mūsā al-Khwārizmī".

Lecture 2: Algorithms

2.1 Algorithm is a procedure for solving problems. specifies a series of steps that perform a particular computation or task. In order to solve a mathematical or computer problem. An algorithm includes calculations, reasoning and data processing. Algorithms can be presented by natural languages, pseudo code and flowcharts, etc.

Algorithm Properties

1. An algorithm is an unambiguous description that makes clear what has to be implemented.
2. A possible way to solve a given problem
3. A mandatory first step before implementing a solution
4. An algorithm expects a defined set of inputs.
5. An algorithm produces a defined set of outputs.
6. An algorithm is guaranteed to terminate and produce a result, always stopping after a finite time. If an algorithm could potentially run forever, it wouldn't be very useful because you might never get an answer.

1. Creating an Algorithm

To begin with we will look at three methods used in creating an algorithm, these are : **STEPPING, LOOPING, CHOOSING**

STEPPING: This is where all the instructions needed to solve our problem are set out one after the other. Here are some examples:

PROBLEM: To find the sum of two numbers.

ALGORITHM:

1. add the two numbers together
2. write down the answer.

PROBLEM: To find the multiply of two number.

ALGORITHM:

1. Multiply the two numbers
2. write down the answer.

Example : Write an algorithm and draw a flowchart that will read the two sides of a rectangle and calculate its area.

Input: Width, Length

Output: Area

Algorithm

Step 1: Input W, L

Step 2: Area = L x W

Step 3: Print Area

GOING LOOPING

if we have to repeat an instruction or a set of instructions a number of times to find a solution for specific problem? In this case we can use loops. We will look at three different types of loops:-

- the REPEAT UNTIL loop
- the WHILE loop
- the FOR loop

PROBLEM: To print the numbers from 10 to 20

ALGORITHM:

For number = 10 to 20

Print number

END FOR

CHOOSING [IF, THEN and ELSE]

we will look at a method for making a choice or a decision. This technique is called CHOOSING, and uses the commands IF, THEN and sometimes (but not always) ELSE.

With this type of command a condition is given with the IF command followed by an action to be taken. If the condition is satisfied, we follow it by the THEN command. A couple of examples which should make things a little clearer:-

Example: Write an algorithm to determine a student's final grade and indicate whether it is passing or failing. The final grade is calculated as the average of four marks.

Input: M1, M2, M3, M4

Output: Grade

Algorithm

Step 1: Input M1,M2,M3,M4

Step 2: GRADE = (M1+M2+M3+M4)/4

Step 3: If (GRADE < 50) then

Print "FAIL"

else

Print "PASS"

End If

Example:

Problem: Given a list of positive numbers, return the largest number on the list.

Inputs: A list L of positive numbers. (This list must contain at least one number).

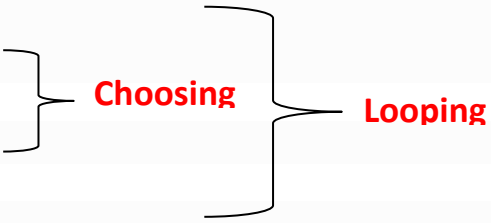
Outputs: A number n, which will be the largest number of the list.

Algorithm:

1. Set max to 0.
2. For each number x in the list L,
3. If x is larger than max, then

   set max to x.

4. End For
5. Print max is now set to the largest number in the list.



Exercises

Determine and Output Whether Number N is Even or Odd

Algorithm:

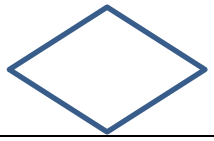
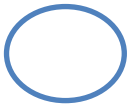
1. Read number N,
2. Set remainder as N modulo 2,
3. If remainder is equal to 0 then
   number N is even,
   else number N is odd,
4. End If

2.2 FLOWCHART

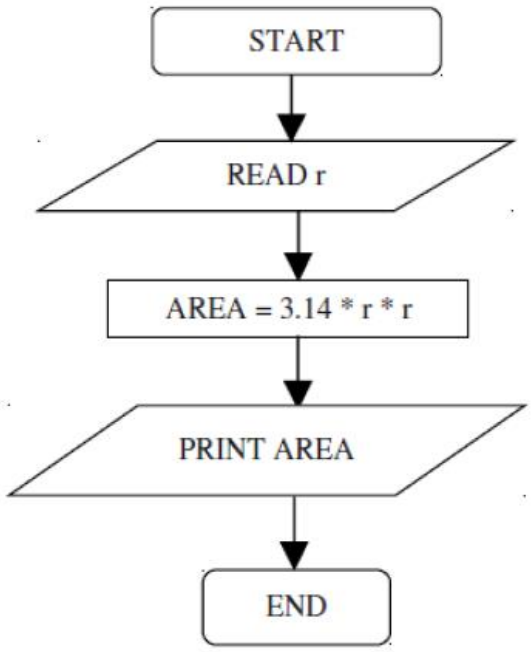
The flowchart is a diagram which visually presents the flow of data through processing systems. This means by seeing a flow chart one can know the operations performed and the sequence of these operations in a system. Algorithms are nothing but sequence of steps for solving problems. So a flow chart can be used for representing an algorithm.

2.2.1 Flowchart Symbols

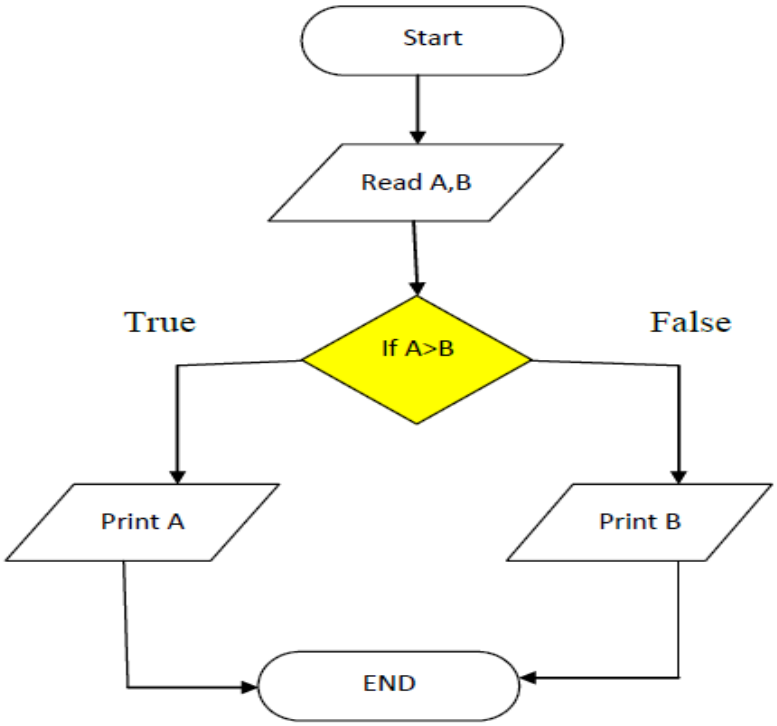
The basic symbols commonly used in flowchart drawing in Programs are: Process, input/output, Decision, Connector and Flow Lines, described as follows:

Symbol	Function
	starting or ending of the program
	Indicates any type of internal operation inside the Processor or Memory
	Used for any Input / Output (I/O) operation. Indicates that the computer is to obtain data or output results.
	Used to ask a question that can be answered in a binary format (Yes/No, True/False)
	Used for connection,
	Shows direction of flow.

Example: draw a flowchart to Find the area of a circle of radius r.



Example: Draw a flowchart to find the greater number between two numbers .



Exercises:

1. Draw a Flowchart to calculate the average from 25 exam scores.
2. Draw a Flowchart for printing the even numbers between 9 and 100.
3. Write An Algorithm to find the Factorial of given number N (المفكوك).