University of Mosul

College of Computer Sciences and Mathematics

Department of Artificial Intelligence

Algorithms and Structured Programming (I)

الخوارزميات والبرمجة المهيكلة (1)

First stage

Lecture 1

أ.م. بيداء سليمان بهنام

Assist.Prof. Baydaa Sulaiman Bahnam

Lecture I Outline

1. Introduction
2. Procedural Programming Principles
3. Algorithms
4. Flowcharts

Introduction

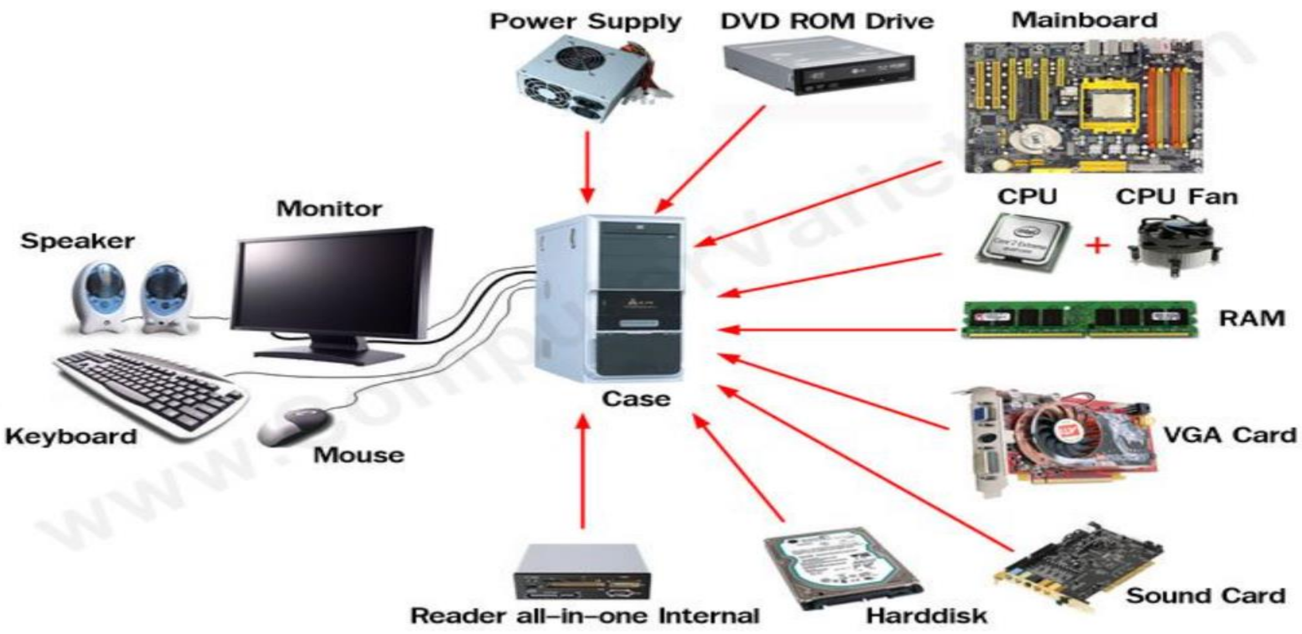
Hardware components

Computer is a device capable of performing computations and making logical decisions at speeds millions and even billions of times faster than human beings.

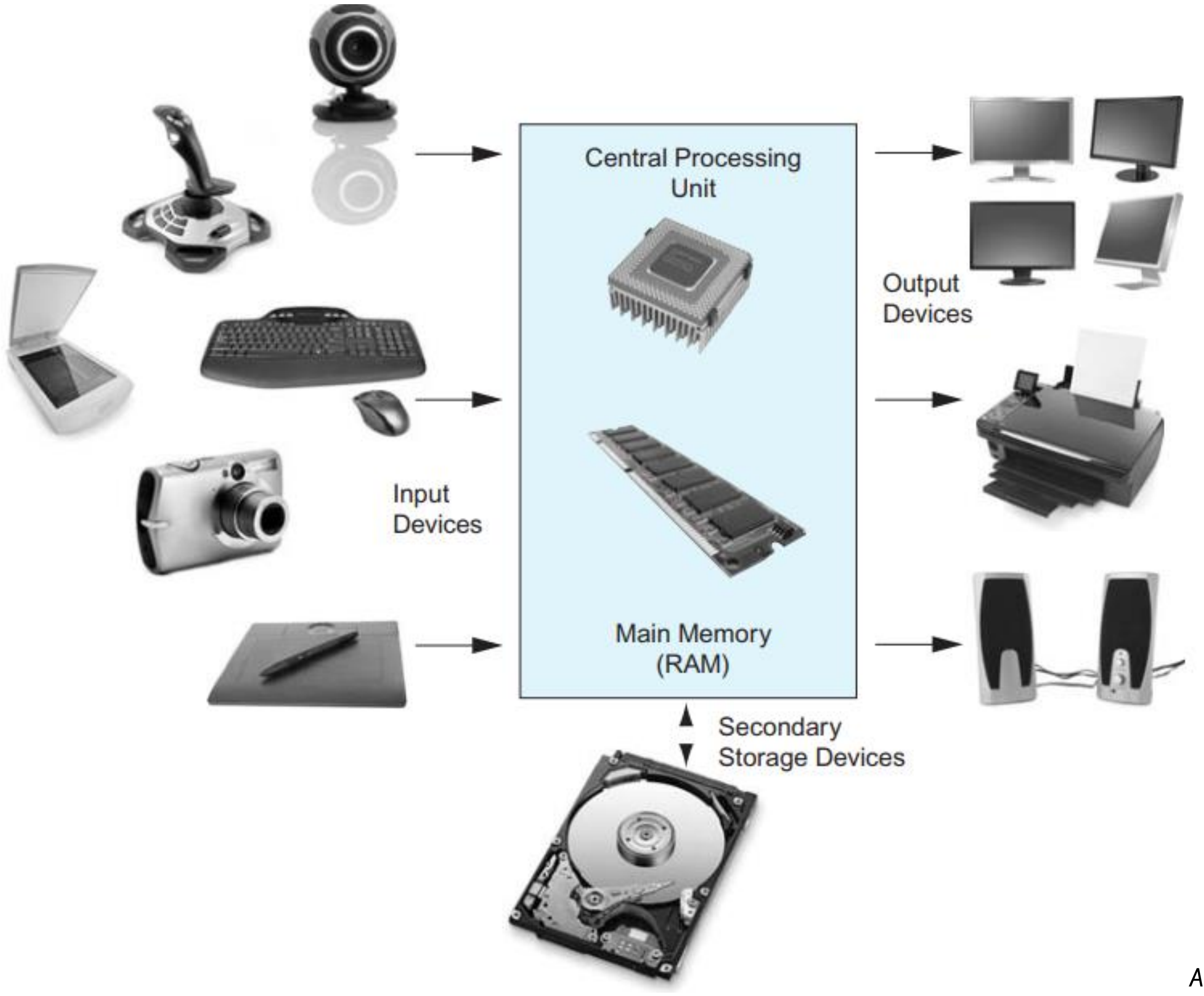
Computers process data under the control of sets of instructions called **computer programs**.

Programming is the process of writing instructions for a computer in a certain order to solve a problem.

The computer programs that run on a computer are referred to as **software (SW)**.While the hard component of it is called **hardware (HW)**.



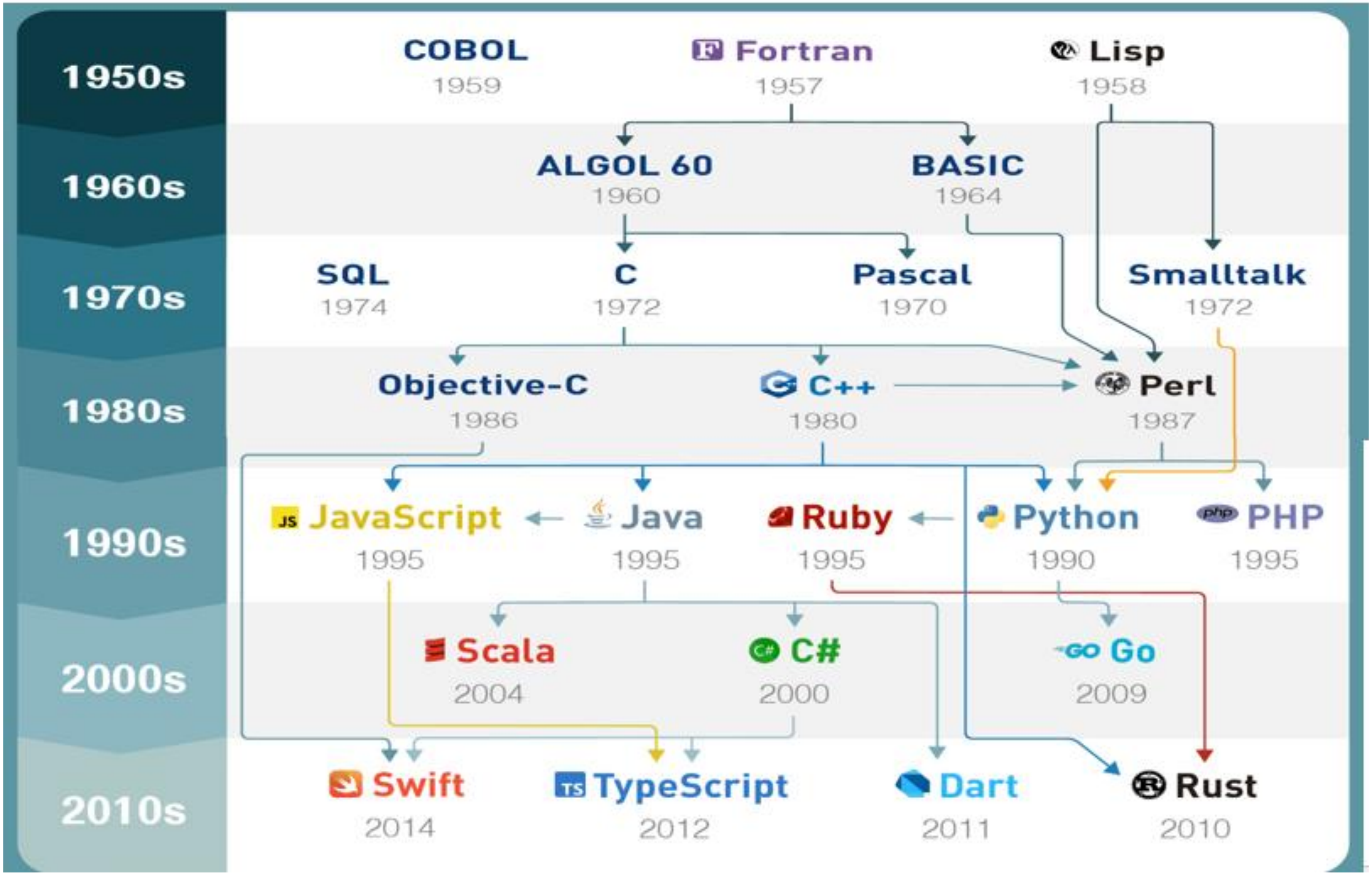
Introduction



A.P. Baydaa Sulaiman

Short history

The following is a short history, just for given a general view of how languages



A procedural programming language

A procedural programming language follows commands sequentially, similar to how computer hardware functions. This relationship results in programs that execute commands efficiently—receiving, interpreting, and executing one instruction at a time. Examples of procedural programming languages include Java, C, Pascal, and BASIC.

Procedural programming follows a set of subroutines, also known as procedures, where the execution of each subroutine happens in a specific order. This makes it easier to reuse code, reduce the amount of repetitive code, and manipulate data. In order for a subroutine to initiate, you have to first call it, and then the program will call the subsequent subroutines.

C++ Programming Language

C++ Programming Language:

For the last couple of decades, the C programming language has been widely accepted for all applications, and is perhaps the most powerful of structured programming languages. Now, C++ has the status of a structured programming language with object-oriented programming (OOP).

C++ has become quite popular due to the following reasons:

1. It supports all features of both structured programming and OOP.

2. C++ focuses on function and class templates for handling data types.

C++ Program Development Process (PDP):

C++ programs typically go through six phases before they can be executed. These phases are:

1. Edit: The programmer types a C++ source program, and makes correction, if necessary. Then file is stored in disk with extension (.cpp).

2. Pre-Processor: Pre-processing is accomplished by the pre-proceccor before compilation, which includes some substitution of files and other directories to be include with the source file.

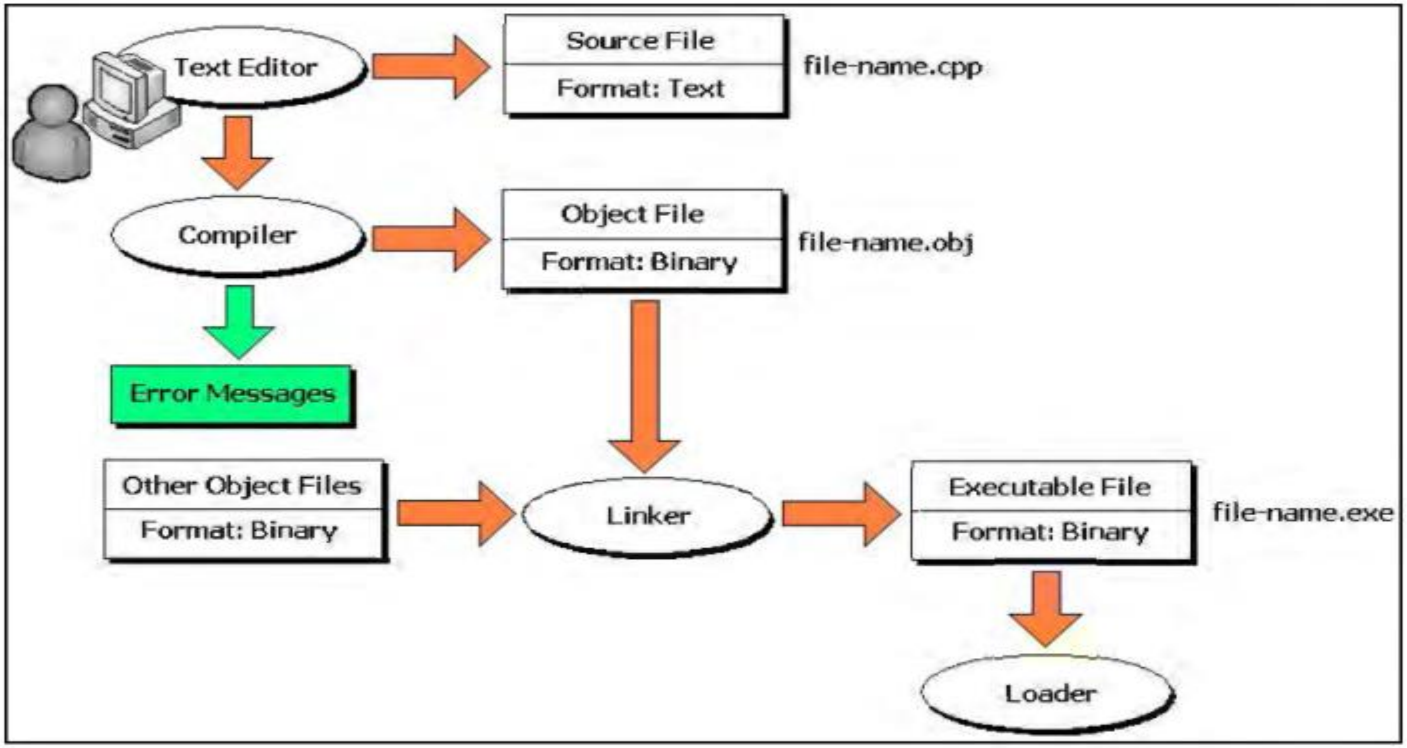
3. Compilation: Converting the source program into object-code.

4. Linking: A linker combines the original code with library functions to produce an executable code.

5. Loading: The loader loads the program from the disk into memory.

6. CPU: Executes the program, residing in memory.

These steps are introduced in the figure below:



Algorithm

Algorithm:

An algorithm can be defined as a finite sequence of effect statements to solve a problem. An effective statement is a clear. unambiguous instruction that can be carried out Thus an algorithm should special the action to be executed and the order in which these actions are to be executed.

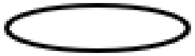
Algorithm properties:

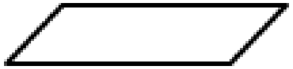
- Finiteness: the algorithm must terminate a finite number of steps.
- Non-ambiguity: each step must be precisely defined. At the completion of each step, the nest step should be uniquely determined.
- Effectiveness: the algorithm should solve the problem in a reasonable amount of time.

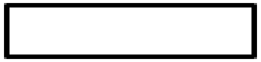
Flowcharts المخططات الانسيابية

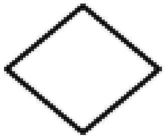
Flowcharts: is a sequence of symbols and boxes with different shapes connected with each other to describe the operations to solve the computer problems .

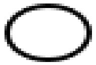
The **symbols** and shapes that used in flowcharts are as follow:

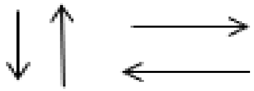
1-  start and end symbol رمز البداية والنهاية

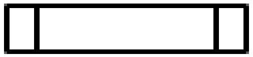
2-  input (read), output (print) symbol رمز الادخال (القراءة) رمز الاخراج (الطباعة)

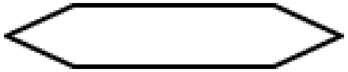
3-  processing symbol, information and arithmetic operation
معالجة الرموز والمعلومات والعمليات الحسابية

4-  decision symbol رمز القرار

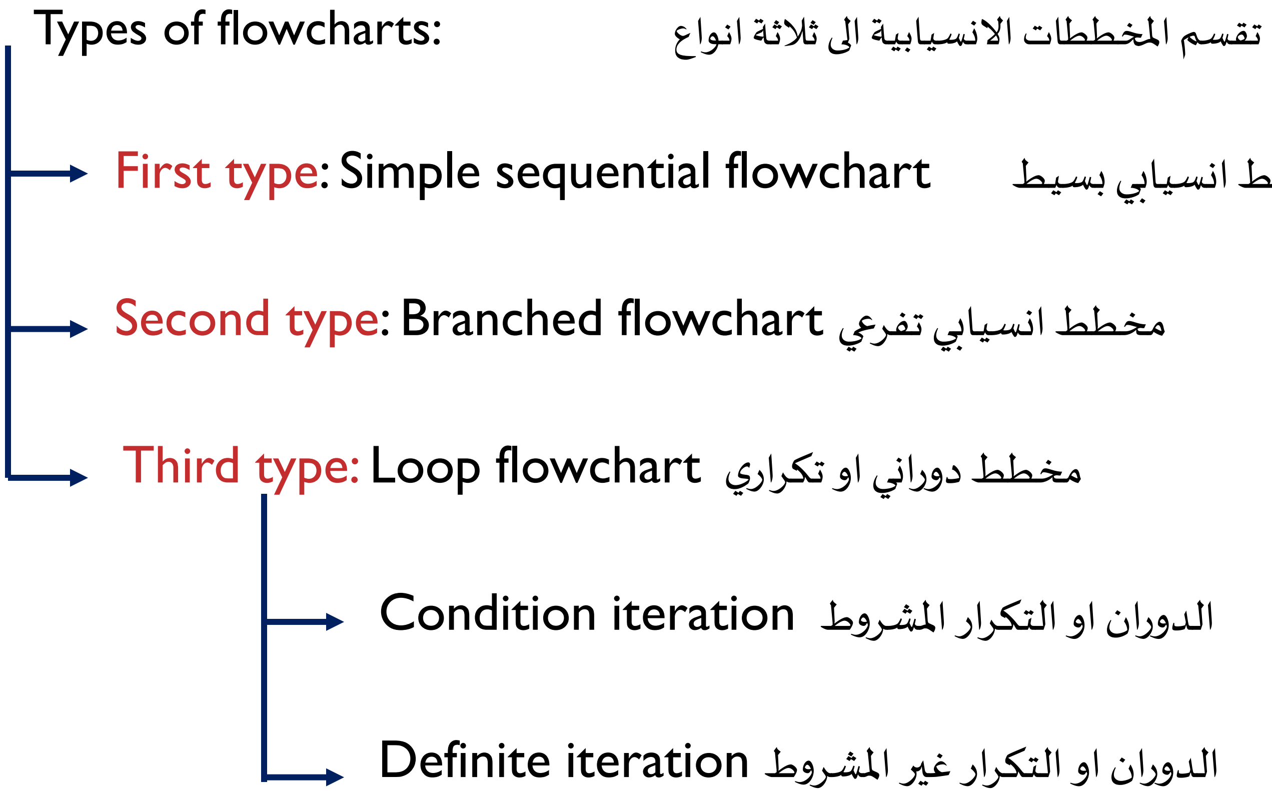
5-  connector symbol رمز التوصيل

6-  flow lines خطوط التدفق

7-  subroutine symbol رمز الروتين الفرعي

8-  loop symbol رمز الحلقة (التكرار)

Types of Flowcharts انواع المخططات الانسيابية



Types of flowcharts: تقسم المخططات الانسيابية الى ثلاثة انواع

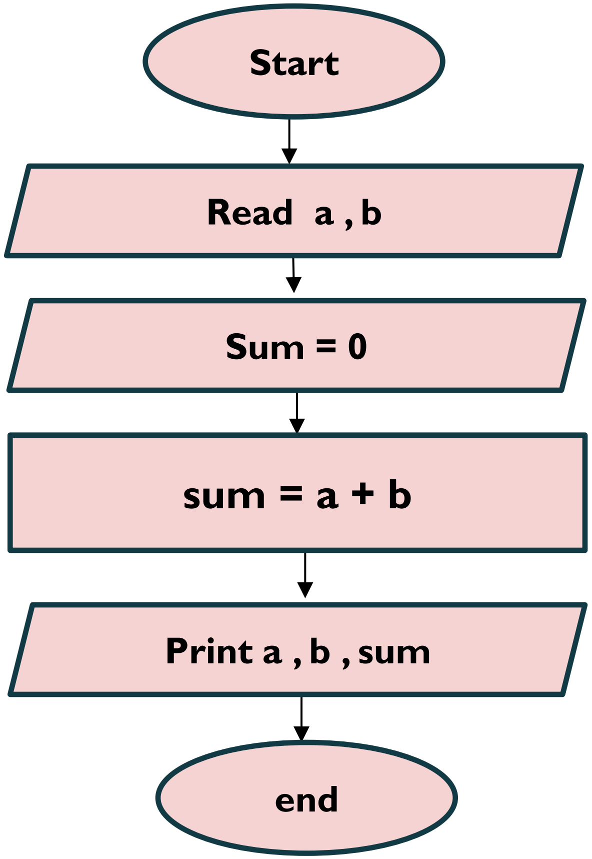
- First type: Simple sequential flowchart مخطط انسيابي بسيط
- Second type: Branched flowchart مخطط انسيابي تفرعي
- Third type: Loop flowchart مخطط دوراني او تكراري
  - Condition iteration الدوران او التكرار المشروط
  - Definite iteration الدوران او التكرار غير المشروط

Examples of simple flowcharts

Ex1: Draw a flowchart to find the sum of two numbers and print them with the result. Then, write algorithm.

(input, read, data, information……..)

(display, print, output, write, result…….)



Algorithm

Input: two numbers a and b

Output: the sum of a and b

Step 1: start

Step 2: read a, b

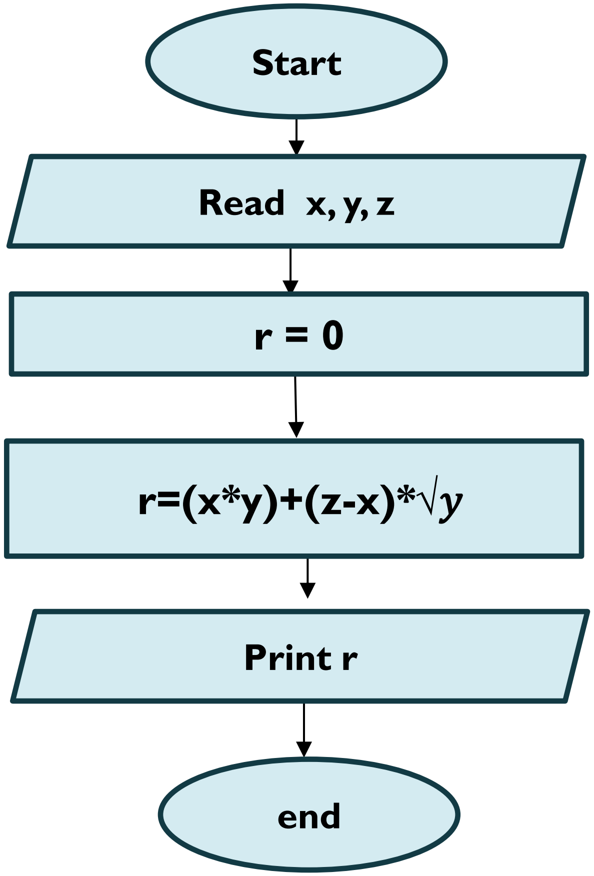
Step 3: sum=0

Step 4: sum = a+b

Step 5 : print a,b, sum

Step 6: stop

Ex2: Draw a flowchart to find r=(x*y)+(z-x)*$\sqrt{y}$ and print the result . Then, write algorithm.



Algorithm

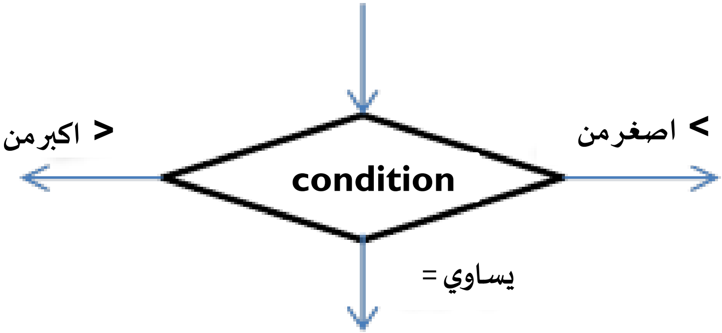
Input: three numbers x, y and z

Output: the result r

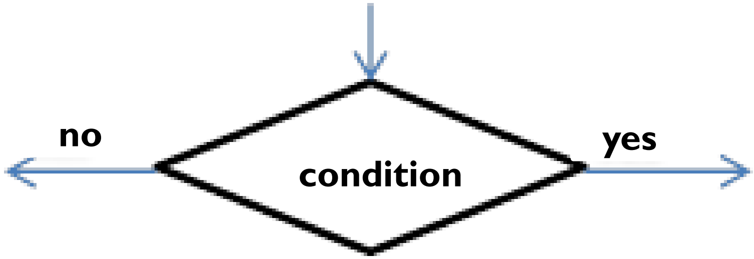
```
Step 1:  start
Step 2:  read x, y, z
Step 3:  r=0
Step 4:  r=(x*y)+(z-x)*$\sqrt{y}$
Step 5 : print r
Step 6:  stop
```

Examples of condition(branched) flowcharts

This symbol is used for making the decision ,it’s had a one input and 3 output. Making the decision depend on the result of the arithmetic operation inside the symbol.



This symbol is used to make the logical decision, yes or no, depending on the result of the logical operation inside the symbol.



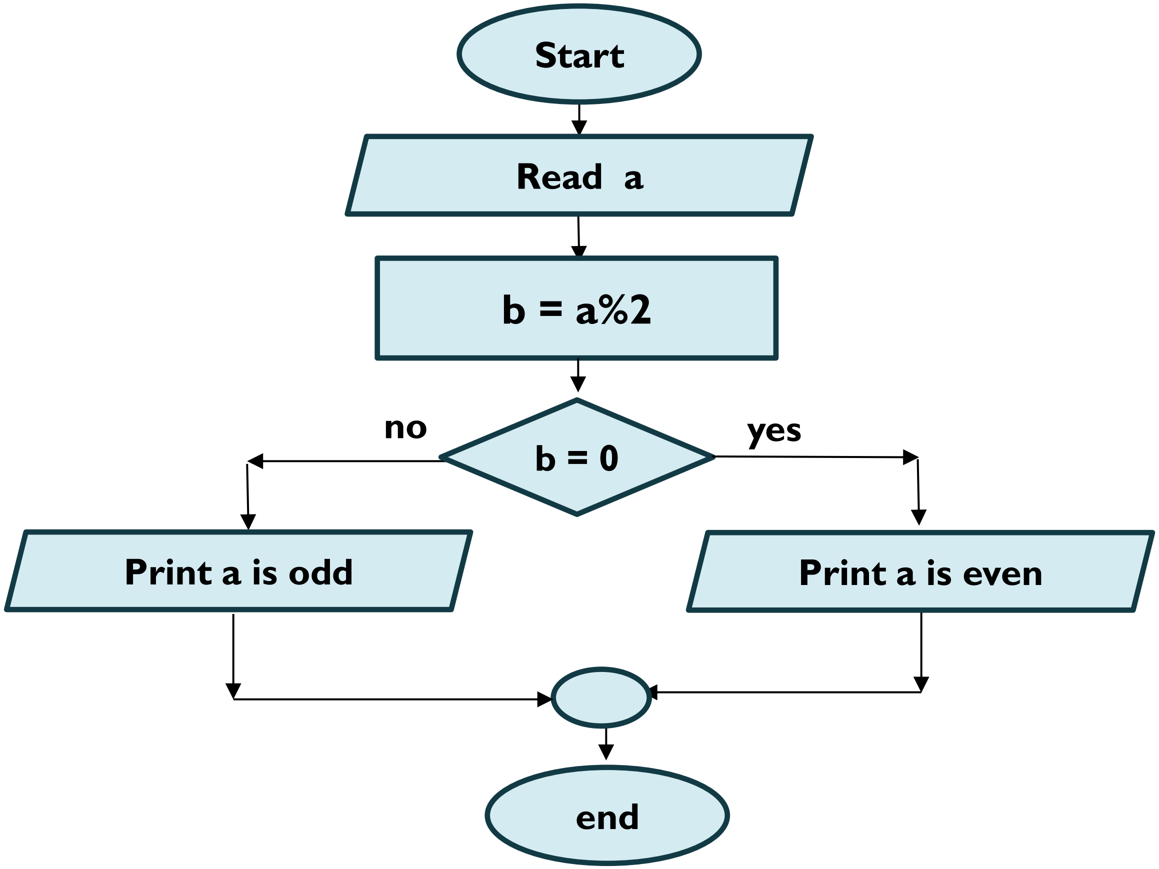
Ex. Draw a flowchart and write an algorithm to read a number and check if it is odd or even. Then, write algorithm.

Algorithm

Input: A number **a**

Output: print **a** is odd or a is even

```
Step 1: start
Step 2: Read a
Step 3: b= a%2
Step 4:if b=0 then
          print a is even number
          else
           print a is odd number
Step 5: stop
```



Ex: Draw a flowchart that reads a number x and calculates the value of z according to the following conditions:

1.if $x<0$ then $z=x+5$

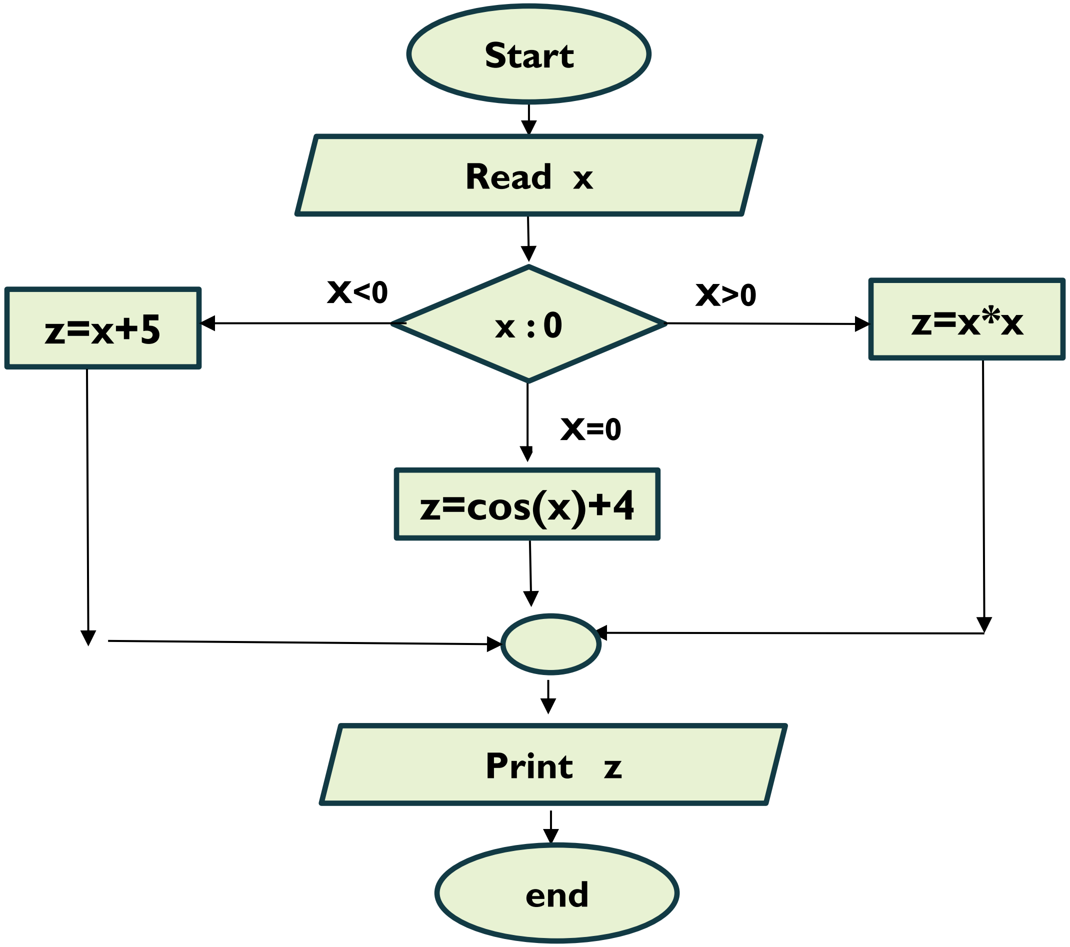
2.if $x=0$ then $z=\cos(x)+4$

3.if $x>0$ then $z=x^2$

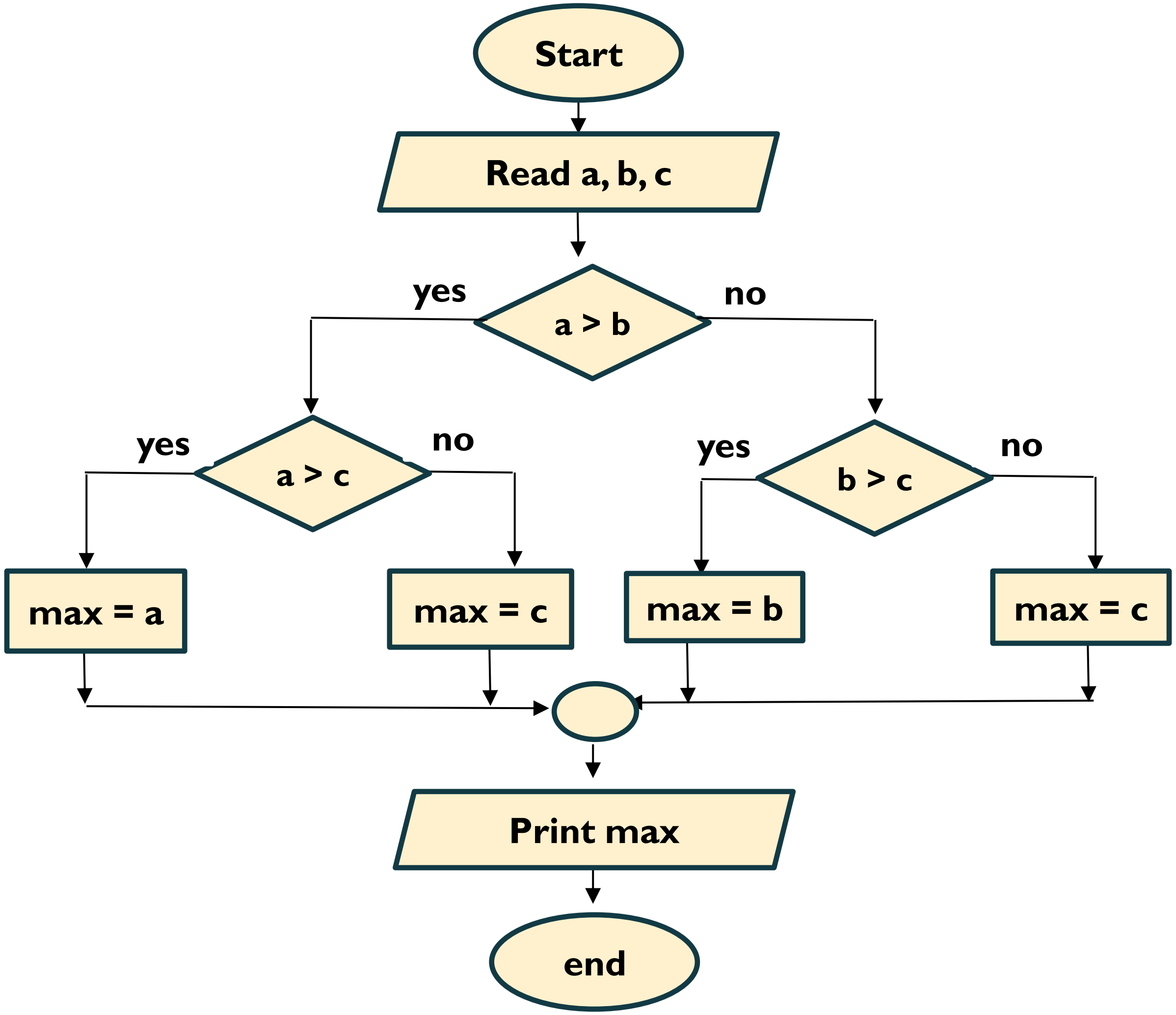
Then, write algorithm.

Algorithm

```
Input: A number x
Output: The value of z
Step 1: Start
Step 2: Read x
Step 3: If x < 0 then
            z = x + 5
Step 4: If x = 0 then
            z = cos(x) + 4
Step 5: If x > 0 then
            z = x * x
Step 6: Print z
Step 7: Stop
```



Ex. Draw a flowchart to find the maximum number of three numbers a, b and c. Then, write algorithm.



Algorithm

Input: Numbers a, b, c

Output: The maximum value max

```
Step 1: Start
Step 2: Read a, b, c
Step 3: If a ≥ b and a ≥ c then
          max = a
Step 4: If b ≥ a and b ≥ c then
          max = b
Step 5: If c ≥ a and c ≥ b then
          max = c
Step 6: Print max
Step 7: Stop
```

A.P. Baydaa Sulaiman