M110: Python Programming

Meeting #1

Algorithms
Flowcharts & Pseudocodes

AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon

Prepared by Dr. Ahmad Mikati

Contents

AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon

Introduction

- Throughout history, man has thought of ever more elegant ways of reducing the amount of labor needed to do things.
- A computer has immense potential for saving time/energy, as most (computational) tasks that are repetitive or can be generalized can be done by a computer.
- For a computer to perform a desired task, a method for carrying out some sequence of events, resulting in accomplishing the task, must be described to the computer.

➡ An **algorithm**.

AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon

Introduction

- Good, logical programming is developed through good pre-code planning and organization (Algorithm) using:
  - Pseudocode
  - Flowcharts
- In this tutorial, you will learn two different ways of laying out a computer algorithm independent of programming language

Algorithm

- An **algorithm** is an **ordered** set of **unambiguous** steps that describes a **process**.
- An algorithm can be implemented in more than one programming language.
- Examples from real life:
  - Recipes
  - Project directions - chemistry lab, writing prompt
  - Instruction manual

AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon

Algorithm

Characteristics of an Algorithm

An algorithm should have the following characteristics:

- **Unambiguous** − Algorithm should be clear and unambiguous. Each of its steps (or phases), and their inputs/outputs should be clear and must lead to only one meaning.
- **Input** − An algorithm should have 0 or more well-defined inputs.
- **Output** − An algorithm should have 1 or more well-defined outputs and should match the desired output.
- **Finiteness** − Algorithms must terminate after a finite number of steps.
- **Feasibility** − Should be feasible with the available resources.
- **Independent** − An algorithm should have step-by-step directions, which should be independent of any programming code.

Algorithm

How to Write an Algorithm

1. **Define the problem**: State the problem you are trying to solve in *clear* and *concise* terms.
2. **List the *inputs*** (information needed to solve the problem) and the *outputs* (what the algorithm will produce as a result)
3. **Describe the steps** needed to convert or manipulate the inputs to produce the outputs. Start at a high level first and keep refining the steps until they are *effectively computable* operations.
4. **Test the algorithm**: choose data sets and verify that your algorithm works!

AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon

Programming Tools

- **Algorithms** are implemented using programming languages.

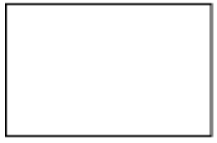
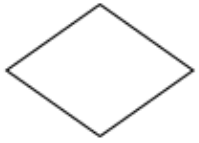
- Programs are written in a programming language such as Java or Python, which is then converted into machine code for the computer to run.

- Algorithms can be designed using **flowcharts** or **pseudocodes**.
  - **Flowchart** - Graphically depicts the logical steps to carry out a task and shows how the steps relate to each other.
  - **Pseudocode** - Uses English-like phrases to outline the program.

Flowcharts

- Flowcharting is a tool developed in the computer industry, for showing the operation of an algorithm.
- A flowchart is a diagram made up of shapes (boxes, diamonds,…) connected by arrows:
  - Each shape represents a step in the process.
  - Arrows show the order in which they occur.

Flowchart symbols

Symbol	Name	Meaning
	*Flowline*	Used to connect symbols and indicate the flow of logic.
	*Terminal*	Used to represent the beginning (Start) or the end (End) of a task.
	*Input/Output*	Used for input and output operations, such as reading and displaying. The data to be read or displayed are described inside.
	*Processing*	Used for arithmetic and data-manipulation operations. The instructions are listed inside the symbol.
	*Decision*	Used for any logic or comparison operations. Unlike the input/ouput and processing symbols, which have one entry and one exit flowline, the decision symbol has one entry and two exit paths. The path chosen depends on whether the answer to a question is "yes" or "no."
	*Connector*	Used to join different flowlines.

Flowcharts

General Rules for flowcharting

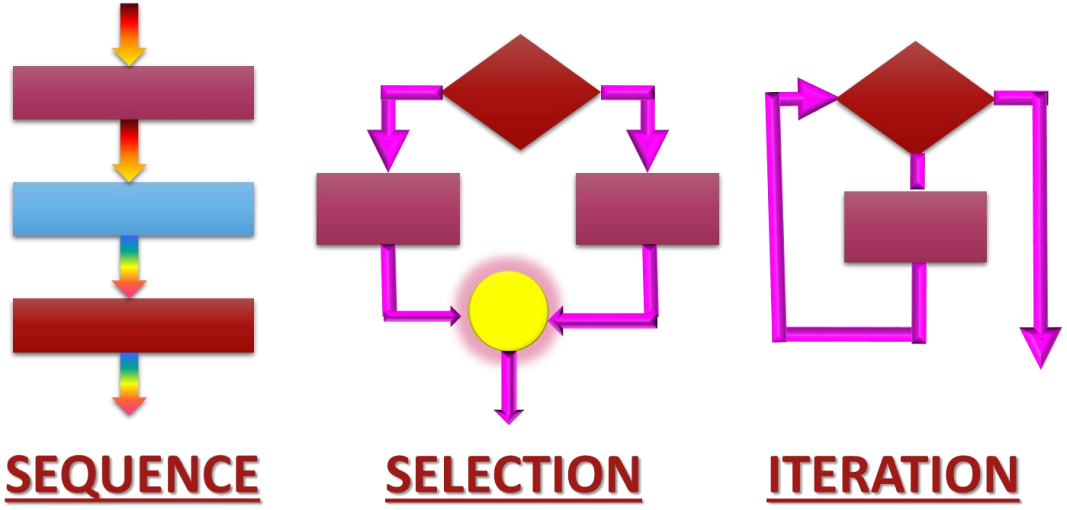
1. All boxes of the flowchart are connected with Arrows (Not lines).
2. Flowchart symbols have an entry point on the top of the symbol with no other entry points. The exit point for all flowchart symbols is on the bottom except for the Decision symbol.
3. The Decision symbol has two exit points; these can be on the sides or the bottom and one side.
4. Generally, a flowchart will flow from top to bottom. However, an upward flow can be shown as long as it does not exceed 3 symbols.
5. Connectors are used to connect breaks in the flowchart. Examples are:
   - From one page to another page.
   - From the bottom of the page to the top of the same page.
   - An upward flow of more then 3 symbols
6. Subroutines and Interrupt programs have their own and independent flowcharts.
7. All flow charts start and end with a Terminal symbol.

Pseudocode

- **Pseudocode** is one of the tools that can be used to write a preliminary plan that can be developed into a computer program.
- Pseudocode is a generic way of describing an algorithm without use of any specific programming language syntax.
- It is pseudo code, it cannot be executed on a real computer, but it models and resembles real programming code and is written at roughly the same level of detail.
- In the algorithm design, the steps of the algorithm are written in free English text, and they may be as long as needed to describe the particular operation.
- Many languages, such as Pascal, Python, have a syntax that is almost identical to pseudocode and hence make the transition from design to coding extremely easy.

Statement structures

- **Sequence** – follow instructions from one line to the next without skipping over any lines
- **Decision** - if the answer to a question is "Yes" then one group of instructions is executed. If the answer is "No," then another is executed
- **Looping** – a series of instructions are executed over and over

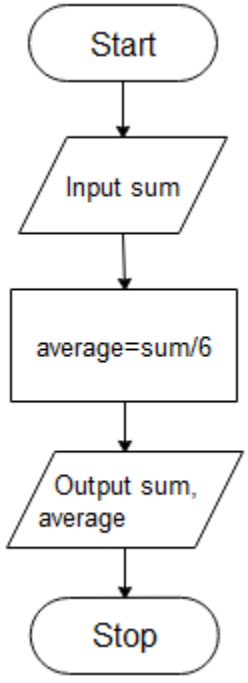


The Sequence Structure

The sequence structure is a case where the steps in an algorithm are constructed in such a way that, no condition step is required.

For example, suppose you are required to design an algorithm for finding the average of six numbers, and the sum of the numbers is given.
The pseudocode and its corresponding flowchart will be as follows:

Start
Use variables sum, average
Input sum
average = sum / 6
Output the sum, average
Stop



The Sequence Structure

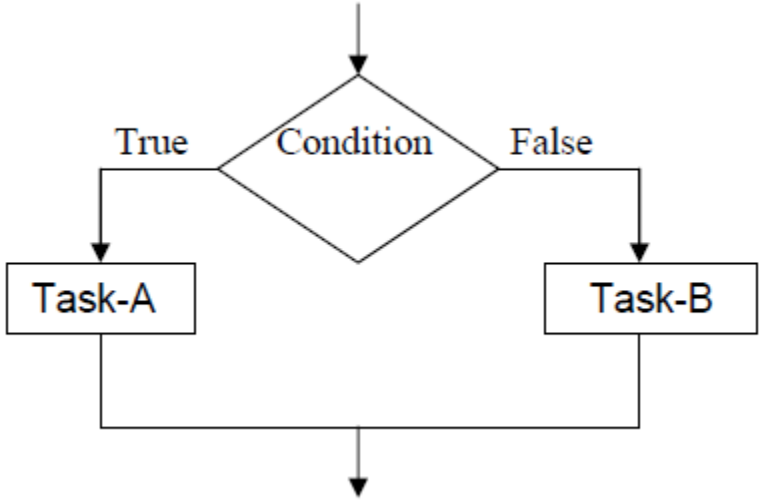
The following pseudo-code describes an algorithm which will accept two numbers from the keyboard and calculate the sum and product Printing the answer on the monitor screen.

```
Start
Use variables sum, product, number1, number2
Input number1, number2
sum = number1 + number2
print sum
product = number1 * number2
print product
Stop
```

AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon

The Decision Structure

- The decision (selection) structure is case where in the algorithm, one has to make a choice of two alternatives by making decision depending on a given condition.
- Selection structures are also called case selection structures when there are two or more alternatives to choose from.



In pseudocode form we get

IF *condition is true*

task A

ELSE

task-B

The Decision Structure

Example:

Write the pseudocode of a program that reads from the user an integer and displays on the screen a message if this integer is odd or even.

Solution:

```
Start
 Use variable: number
Input number
IF number mod 2 = 0
    print ("The number is even")
ELSE
   print ("The number is odd")
Stop
```

The Decision Structure

Example:

Write the pseudocode to do the following:

- Read from the user two values representing the height and base of a triangle.
- Check if the user entered positive values (i.e. greater than 0), then calculate and print the area of the triangle. Otherwise, print a message that the values should be positive and do not calculate that area.

The area of triangle is calculated as: ½ (base)(height)

The Decision Structure

Solution:

```
Start
 Use variable: base, height, area
Input base
Input height
IF base >0 and height >0
    area = ½ (base)(height)
    print (area)
ELSE
   print ("You should enter positive numbers")
Stop
```

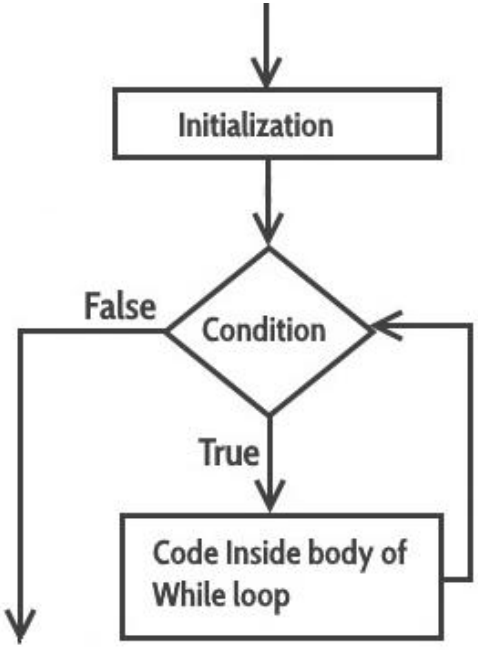
AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon

Repetition

Repetition or Iteration Structure

- Any program instruction that repeats some statement or sequence of statements several times is called an **iteration** or a **loop**.
- The commands used to create iterations or loops are all based on **logical tests**.
- The repetition structure can be done with the while loop.
- **The While loop**: is used to repeat a statement or a set of statements as long as a condition is true.
- The pseudocode syntax and flowchart of the while loop are:

WHILE (condition)
A statement or block of statements
ENDWHILE



Repetition

Example:

Write the pseudocode for reading the values of 6 test scores and finding their sum.

Solution 1: (impractical!)

a) Pseudocode

1. Start
2. sum = 0
3. Get the 1st testscore
4. Add first testscore to sum
5. Get the 2nd testscore
6. Add to sum
7. Get the 3rd testscore
8. Add to sum
9. Get the 4th testscore
10. Add to sum
11. Get the 5th testscore
12. Add to sum
13. Get the 6th testscore
14. Add to sum
15. Output the sum
16. Stop

- Notice that there are repeated steps in the solution.

  You should use a loop.

AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon

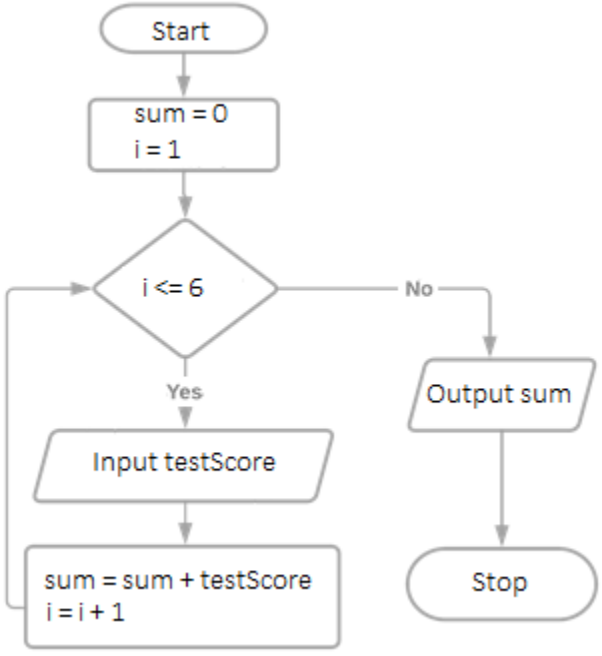
Repetition

Solution 2:

a) Pseudocode

```
Start
 Use variable: testScore, sum, i
sum= 0
i = 1
WHILE( i<=6)
      Input testScore
      sum = sum + testScore
      i = i + 1
ENDWHILE
Output sum
Stop
```

b) Flowchart



Repetition

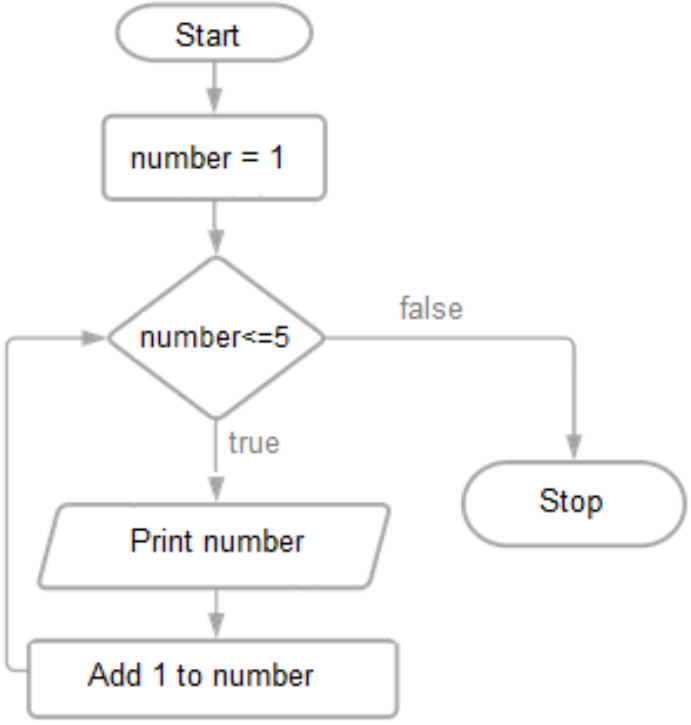
Example:
Write a pseudo-code and a flowchart for a program that prints the numbers from 1 to 5.

Solution:

a) Pseudocode

```
Use variable: number
number = 1
WHILE number <= 5
        Print number
        Add 1 to number
ENDWHILE
```

b) Flowchart



Repetition

Example:

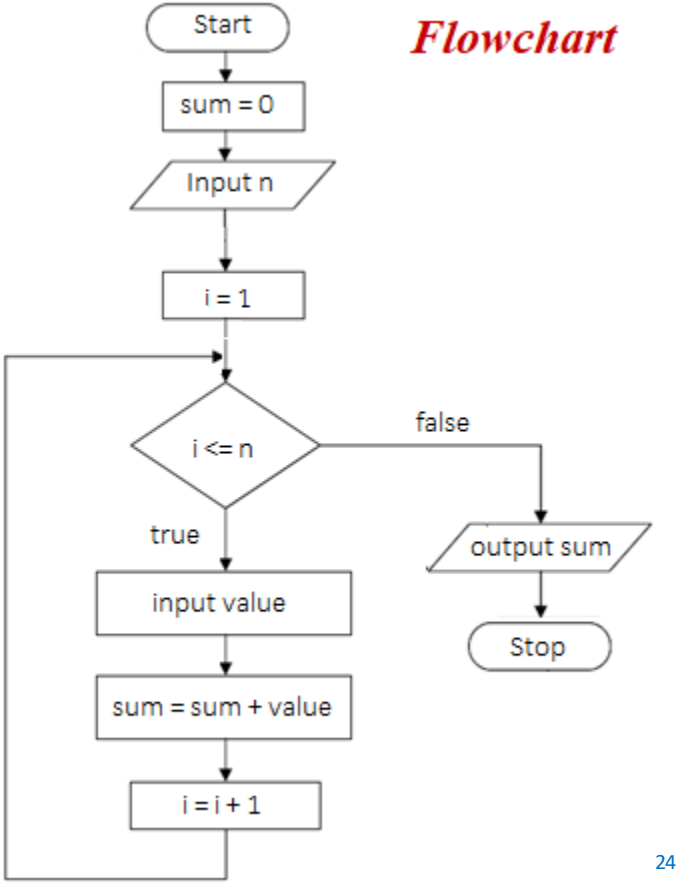
Design a pseudocode and the corresponding flowchart for finding the sum of **n** numbers, where n is a value to read from the user.

Solution:

Pseudocode

```
Start
Use variable: value, sum, n, i
sum = 0
i = 1
Input n
WHILE (i <= n)
      Input value
      sum = sum + value
       i= i+1
ENDWHILE
Output sum
Stop
```

Flowchart



Extra Examples on Pseudocodes and Flowcharts

AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon

Example 1

Write a pseudocode to print out each character typed at a keyboard until the character 'q' is entered. (i.e., The program reads characters and print them until 'q' in entered).

Problem explanation:

- This is an example of **sentinel-controlled** repetition (indefinite repetition) because the number of repetitions is not known before the loop begins executing.
- A special value called a **sentinel value** can be used to indicate "end of data entry" or a "condition of termination".
- A sentinel value must be chosen that cannot be confused with an acceptable input value.

Example 1- **Solution**

Pseudocode

```
Use variable: letter
Print "Type in a character or 'q' to stop"
Input letter
WHILE letter <> 'q'
        Print letter
        Print "Type in a character or 'q' to stop"
        Input letter
ENDWHILE
```

AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon

Example 2

Design a pseudocode that will output the square of any number input until the number input is zero.

Solution:

```
Use variable: number, square
Print "Type in a number or zero to stop"
Input number
WHILE number <> 0
        square = number * number
        Print square
        Print "Type in a number or zero to stop"
        Input number
ENDWHILE
```

Example 3

Design the pseudocode and flowchart for a program that reads the grades of several students in a class and calculates the grade-point average for the class. The total number of students is unknown, and the program should stop reading grades when the user enters -1.

Example 3- Solution

Pseudocode

Start

Use variable: counter , grade, sum, average

Initialize counter to 0

Initialize sum to 0

Print "Type in a grade or -1 to stop"

Input grade

WHILE grade ≠ -1

    Add the Grade to the Sum

    Increment the Counter

    Print "Type in a grade or -1 to stop"

    Input grade

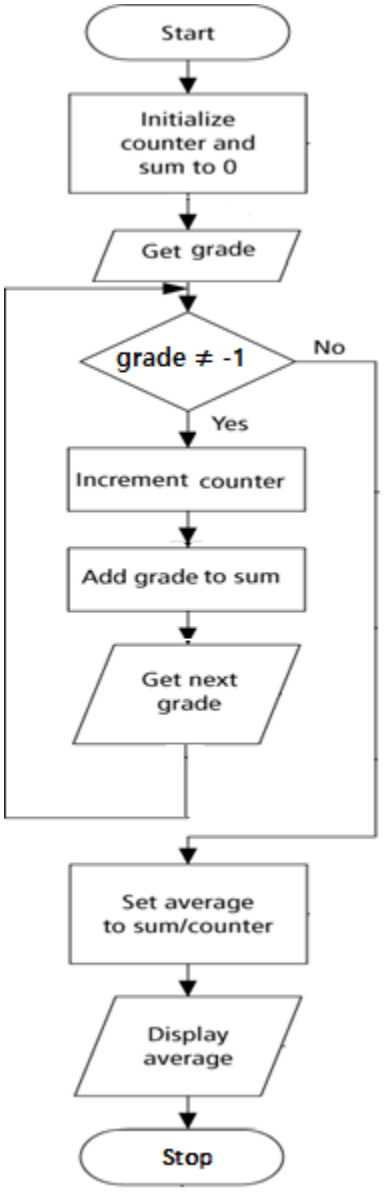
ENDWHILE

Average = Sum / Counter

Print Average

Stop

Flowchart



Summary

- Laying out an algorithm using flowcharts and pseudo-code
- Learning basic elements of algorithms:
  - Input
  - Output
  - Decision-Making
  - Repetition
  - Processes

AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon

Extra Exercises

Write an algorithm (pseudocode or a flowchart) to do the following:

1. Read two numbers from the user. Check if the user entered values greater than Zero, then display the sum of both numbers; otherwise display the multiplication of both numbers. Print "Finish".
2. Read from the user a student's grades in TMA, MTA and final exam. The program should calculate the total grade and displays in the screen if the student passed or not. A student passes a course if the total grade is 50 or more.
3. Read three scores (quiz, midterm, and final). Calculate the total score and determine and print the grade based on the following rules:
   - if the total score >= 90 → grade=A
   - if the total score >= 70 and <90 → grade=B
   - if the total score>=50 and <70 → grade=C
   - if the total score<50 → grade=F

Extra Exercises

4. Print all odd numbers from 1 to 100.
5. Print all even numbers from 1 to 100.
6. Prompts the user to enter 10 positive numbers. Find and print the biggest number entered by the user.
7. Prompt the user to enter positive numbers (0 to stop). find and print the smallest number entered by the user.
8. Read from the user 10 words. Count and print the number of words that start with character 'a'.
9. Read from the user words ("finish" to stop). Count and print the number of words that start with character 'a'.
10. Print the multiplication table for 6:

    1 x 6 = 6

    2 x 6 = 12

    3 x 6 = 18

    …

    12 x 6 = 72

AOU
الجامعة العربية المفتوحة
Arab Open University
Lebanon