SWE1301 : Introduction to Problem Solving and Software Development

Lecture 08 : Flowcharts, Coding & Testing
At: CIT Theatre
12-1pm
By: M .I. Mukhtar

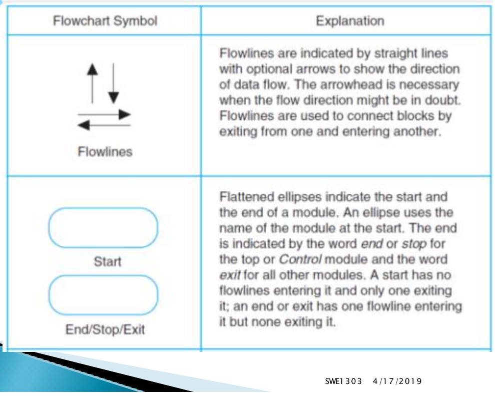
Lecture Outline

- Flowcharts
- Coding
- Testing
- Sequential structures

Flowchart

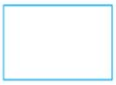
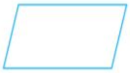
- From the algorithms the programmer develops the **flowcharts**
  - Flowcharts are graphic representations of the algorithms.
  - flowchart shows the flow of the processing from the beginning to the end of a solution.
  - Each block in a flowchart represents one instruction from an algorithm.
  - A flowchart will show errors in logic not readily visible in the other charts.
  - There are flowchart symbols for use with various types of processing.

Flowchart Symbols

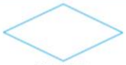
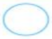
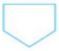


Flowchart Symbol	Explanation
Flowlines	Flowlines are indicated by straight lines with optional arrows to show the direction of data flow. The arrowhead is necessary when the flow direction might be in doubt. Flowlines are used to connect blocks by exiting from one and entering another.
Start End/Stop/Exit	Flattened ellipses indicate the start and the end of a module. An ellipse uses the name of the module at the start. The end is indicated by the word *end* or *stop* for the top or *Control* module and the word *exit* for all other modules. A start has no flowlines entering it and only one exiting it; an end or exit has one flowline entering it but none exiting it.

Flowchart Symbols

Flowchart Symbol	Explanation
 Processing	The rectangle indicates a processing block, for such things as calculations, opening and closing files, and so forth. A processing block has one entrance and one exit.
 I/O	The parallelogram indicates input to and output from the computer memory. An input/output (I/O) block has one entrance and only one exit.

Flowchart Symbols

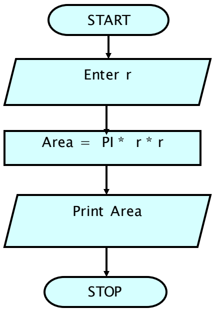
Flowchart Symbol	Explanation
 Decision	The diamond indicates a decision. It has one entrance and two and only two exits from the block. One exit is the action when the resultant is *True* and the other exit is the action when the resultant is *False*.
 On-Page Connectors*  Off-Page Connectors*	Flowchart sections can be connected with two different symbols. The circle connects sections on the same page, and the home base plate connects flowcharts from page to page. Inside these two symbols the programmer writes letters or numbers. The on-page connector uses letters inside the circle to indicate where the adjoining connector is located. An *A* connects to an *A*, a *B* to a *B*, etc. The off-page connectors use the page number where the next part or the previous part of the flowchart is located. This allows the reader to easily follow the flowchart. On- and off-page connectors will have either an entrance or an exit.

Example 6: Flowchart

From Example 1:

Algorithm

- Step 1: Enter r
- Step 2: Area = PI * r * r
- Step 3: Print Area
- Step 4 : End

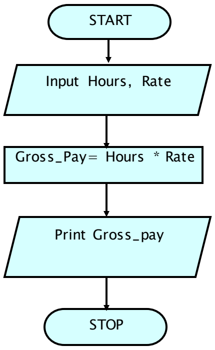


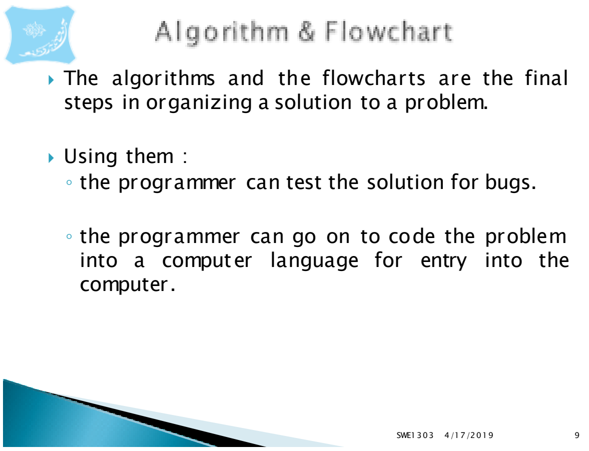
Example 7: Flowchart

From Example 2:

Algorithm

- Step 1: Enter Hours, Rate
- Step 2: Gross_Pay = Hours * Rate
- Step 3: Print Gross_Pay
- Step 4 : End



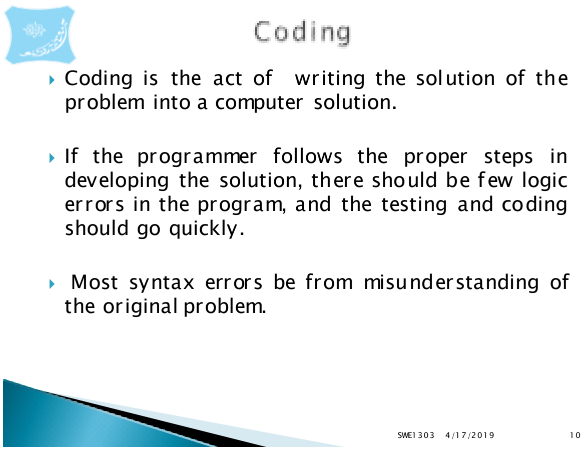


Algorithm & Flowchart

- The algorithms and the flowcharts are the final steps in organizing a solution to a problem.

- Using them :
  - the programmer can test the solution for bugs.

  - the programmer can go on to code the problem into a computer language for entry into the computer.

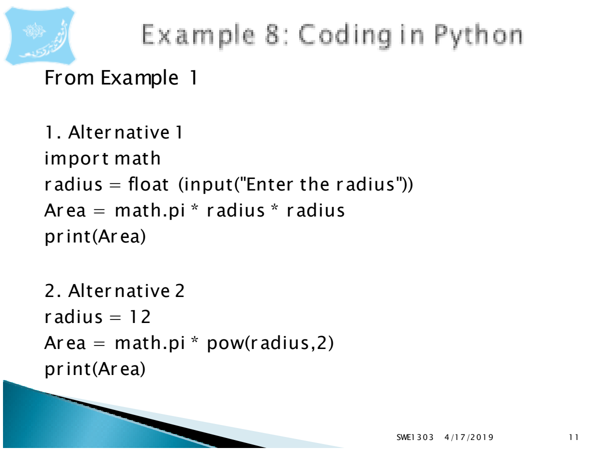


Coding

- Coding is the act of writing the solution of the problem into a computer solution.

- If the programmer follows the proper steps in developing the solution, there should be few logic errors in the program, and the testing and coding should go quickly.

- Most syntax errors be from misunderstanding of the original problem.



Example 8: Coding in Python

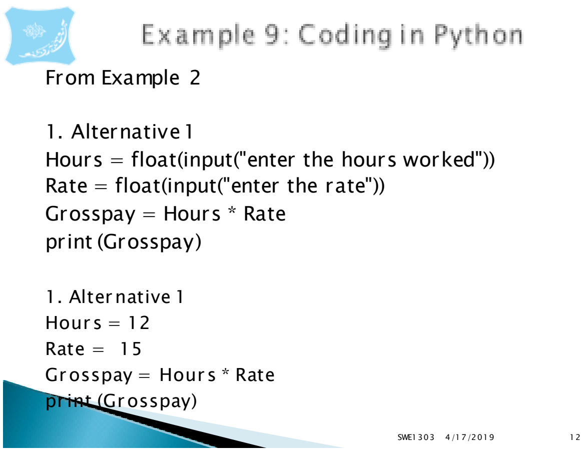
From Example 1

1. Alternative 1

```
import math
radius = float (input("Enter the radius"))
Area = math.pi * radius * radius
print(Area)
```

2. Alternative 2

```
radius = 12
Area = math.pi * pow(radius,2)
print(Area)
```



Example 9: Coding in Python

From Example 2

1. Alternative 1

```
Hours = float(input("enter the hours worked"))
Rate = float(input("enter the rate"))
Grosspay = Hours * Rate
print (Grosspay)
```

1. Alternative 1

```
Hours = 12
Rate =  15
Grosspay = Hours * Rate
print (Grosspay)
```

Testing the Solution

- When a solution is complete, it is important to test it to make sure no errors in logic or in the setup of the expressions and equations.
  - This can be carried out manually before coding
- To test a solution, the programmer selects test data, a set of values for the input data, and works them through every step in the solution.
- It is important to select test data carefully, so the correctness of the results can be checked with as much accuracy as possible

Example 10: Testing in Python

- From Example 9, Alternative 2 Test with the following:
- a. Hours = 15 Rate = 12
- b. Hours = 5 Rate = 2.5

Sequential Logic Structure

Structures of Algorithms

- The **sequential structure** executes instructions one after another in a sequence.
- The **decision structure** branches to execute one of two possible sets of instructions.
- The **loop structure** executes a set of instructions many times.

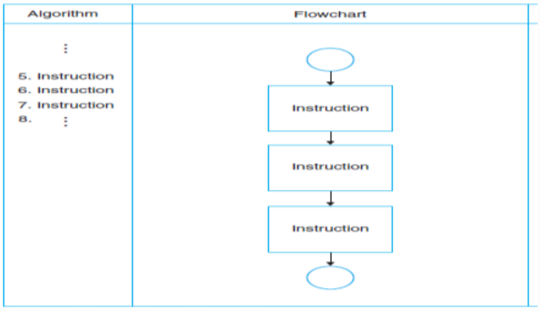
Sequential Logic Structure

- The most commonly used and the simplest logic structure is the sequential structure.
- All problems use the sequential structure, and most problems use it in conjunction with one or more of the other logic structures.

Sequential Logic Structure…

- The **sequential structure** executes instructions one after another in a sequence.
- A programmer who uses the sequential logic structure is asking the computer to :
  - process a set of instructions in sequence from the top to the bottom of an algorithm.

Sequential Logic Structure Format



Algorithm	Flowchart
⋮ 5. Instruction 6. Instruction 7. Instruction 8. ⋮	

Example 1

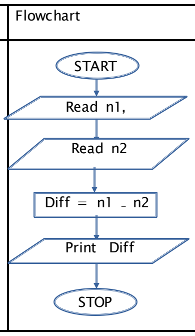
- Write an algorithm and draw the flowchart to read two numbers (n1 and n2) and find their difference.
  - Use PAC
- Desk Check with the following
  - 10 and 16
  - 40 and 32
  - Desk check is a manual technique for checking the logic and correctness of an algorithm.

Solution- PAC

Given Data	Required result
n1 , n2	Difference (Diff)
Required Processing	Solution alternative
Diff = n1 – n2	Define n1 and n2 as a constant or input value.

Solution- Algorithm & Flowcharts

Algorithm	Flowchart	Test

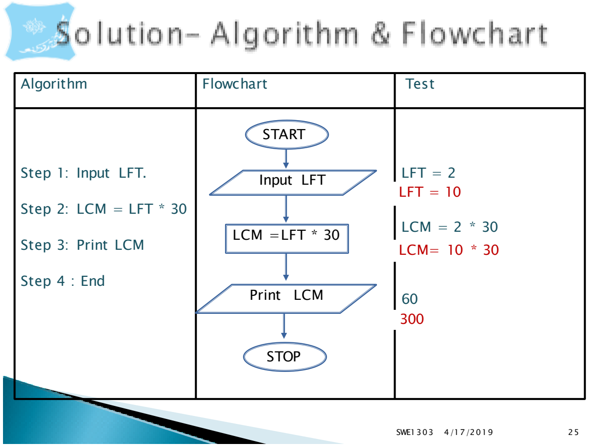
Step 1: Read n1.
Step 2: Read n2.
Step 3: Diff = n1–n2
Step 4: Print Diff
Step 5 : End
	
	START
	Read n1,
	Read n2
	Diff = n1 _ n2
	Print Diff
	STOP
		n1=10, n2= 16
		n1=40, n2 = 32
		Diff = 10 - 16
		Diff= 40 - 32
		-6
		12

Example 1

- Write an algorithm and draw a flowchart to convert the length in feet(LFT) to length in centimeter (LCM) . **Given that LCM= LFT* 30**
  - Use PAC
- Desk Check with the following
  - 2
  - 10

Solution- PAC

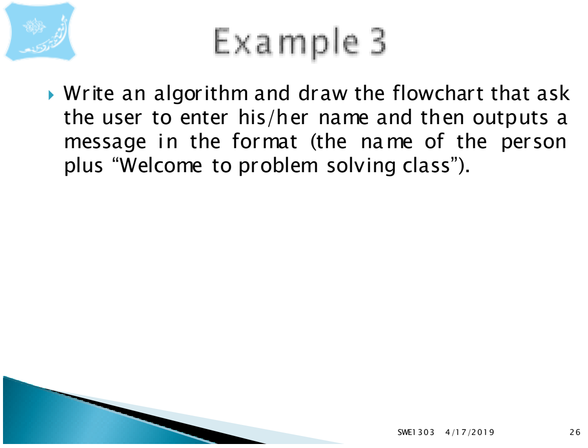
Given Data	Required result
LFT	LCM
Required Processing	Solution alternative
LCM= LFT * 30	Define LCM as a constant or input value.



Solution- Algorithm & Flowchart

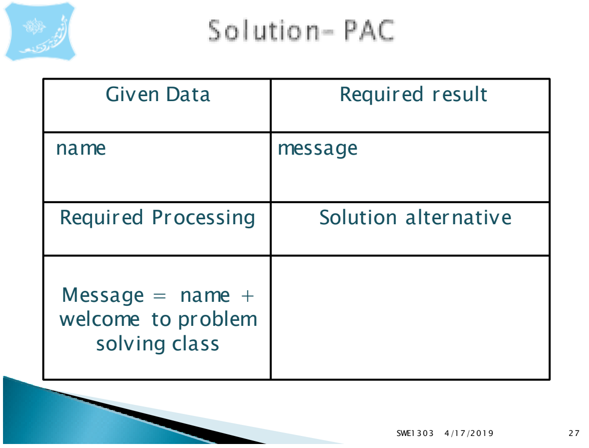
Algorithm	Flowchart	Test

Step 1: Input LFT.
Step 2: LCM = LFT * 30
Step 3: Print LCM
Step 4 : End
	START
	Input LFT
	LCM =LFT * 30
	Print LCM
	STOP
		LFT = 2
		LFT = 10
		LCM = 2 * 30
		LCM= 10 * 30
		60
		300



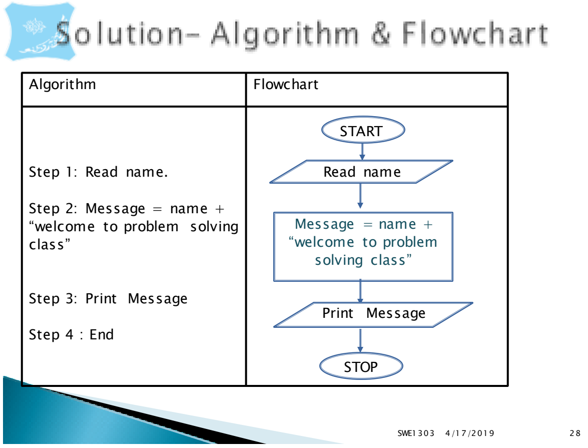
Example 3

- Write an algorithm and draw the flowchart that ask the user to enter his/her name and then outputs a message in the format (the name of the person plus “Welcome to problem solving class”).



Solution- PAC

Given Data	Required result
name	message
Required Processing	Solution alternative
Message = name + welcome to problem solving class	



Solution- Algorithm & Flowchart

Algorithm	Flowchart
Step 1: Read name. Step 2: Message = name + “welcome to problem solving class” Step 3: Print Message Step 4 : End	START Read name Message = name + “welcome to problem solving class” Print Message STOP

Summary

- Certain organizational tools such as PAC, pseudocode, algorithms and Flowcharts will help programmers to learn how to solve problems on the computer.
- The **sequential structure** executes instructions one after another in a sequence.

Quiz 2

- Write an algorithm and draw the flowchart to read two numbers (a and b) and find their product.
  - Use PAC

Questions !!!