



Software Engineering

هندسة البرمجيات

جامعة بغداد

كلية التربية للعلوم الصرفة/ابن

الهيثم

قسم علوم الحاسبات

المرحلة الثالثة

أ.م.د. علي يحيى غني

نظري

محاضرة الاسبوع الرابع عشر

Structured Analysis

(Pseudocode + Flow Chart)

Topics covered



- ✧ Structured Analysis
- ✧ Structured Analysis Models (Tools):
 - i. Data Flow Diagram
 - ii. Data Dictionary
 - iii. Decision Trees
 - iv. Decision Tables
 - v. Pseudocode
 - vi. Flow Chart

Pseudocode



- ✧ Pseudocode is an informal way of programming description that does not require any strict programming language syntax or underlying technology considerations.
- ✧ It is used for creating an outline or a rough draft of a program.
- ✧ Pseudocode summarizes a program's flow, but excludes underlying details.
- ✧ System analyst and designers write pseudocode to ensure that programmers understand a software project's requirements and align code accordingly.

Example of Pseudocode



✧ Conditions:

C1: Advance payment made

C2: Purchase amount = Rs 10,000/-

C3: Regular Customer

✧ Actions:

A1: Give 5% discount

A2: Give no discount

Example of Pseudocode



✧ Rules:

R1: Yes Advance payment made = Give 5% discount

R2: No Advanced payment made + Yes Purchase amount = Rs 10,000/- + Yes Regular Customer = Give 5% discount

R3: No Advanced payment made + Yes Purchase amount = Rs 10,000/- + No Regular Customer = Give no discount

R4: No Advanced payment made + No Purchase amount = Rs 10,000/- = Give no discount

Ru1 +R2 = Give 5% discount R3+R4 = Give no discount

Example of Pseudocode



```
if customer pays advance
then
    Give 5% Discount
else
    if purchase amount >=10,000
    then
        if the customer is a regular customer
        then Give 5% Discount
        else No Discount
        end if
    else No Discount
    end if
end if
```

Figure 19: Pseudocode

Flow Chart



- ✧ A flowchart is a graphical representations of steps. It was originated from computer science as a **tool for representing algorithms and programming logic** but had extended to use in all other kinds of processes.
- ✧ Nowadays, flowcharts play an extremely important role in displaying information and assisting reasoning.
- ✧ They help us visualize complex processes, or make explicit the structure of problems and tasks.

Flow Chart Symbols

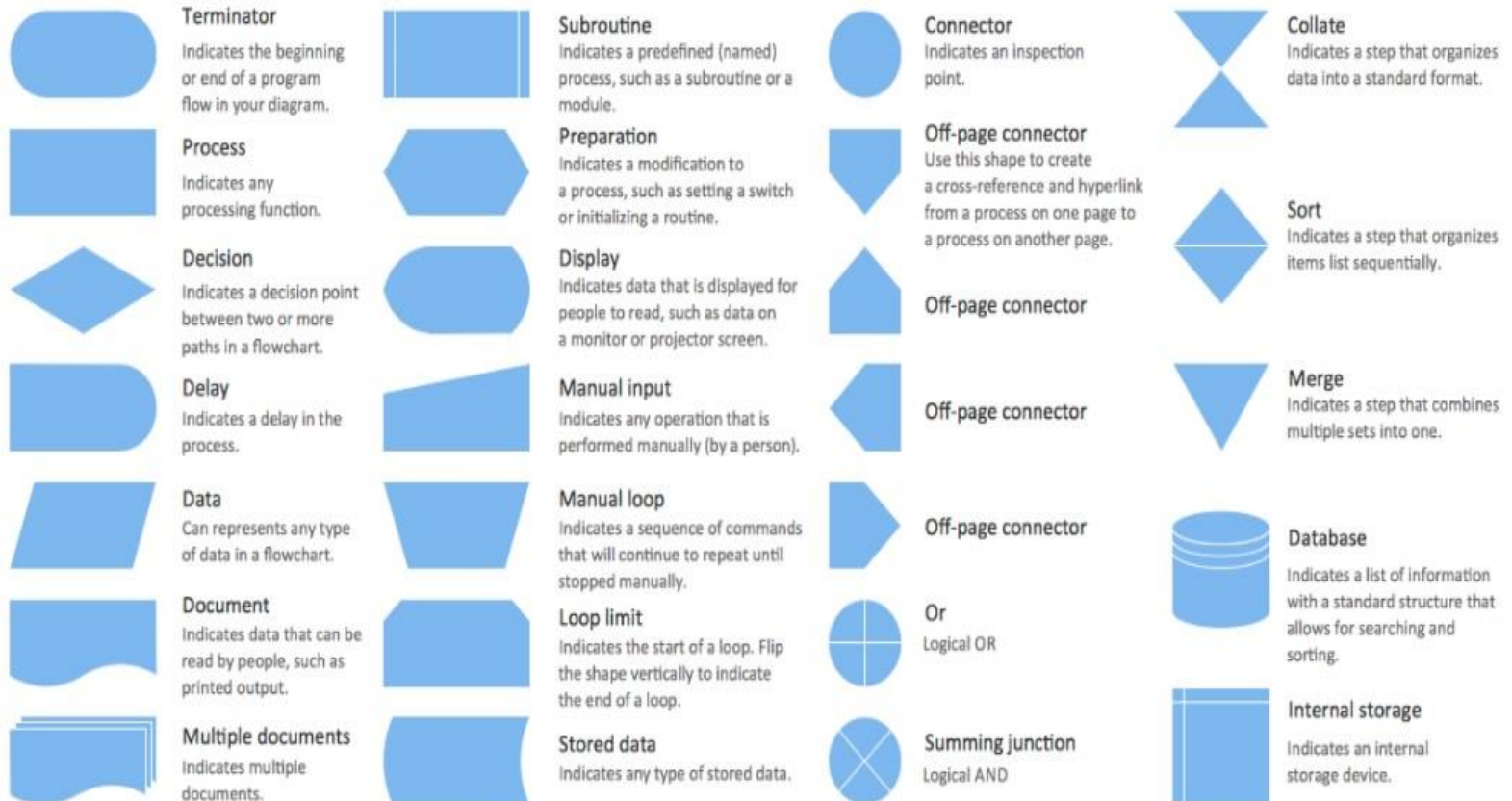


Figure 20: Flow Chart Symbols

Pseudocode & Flow Chart



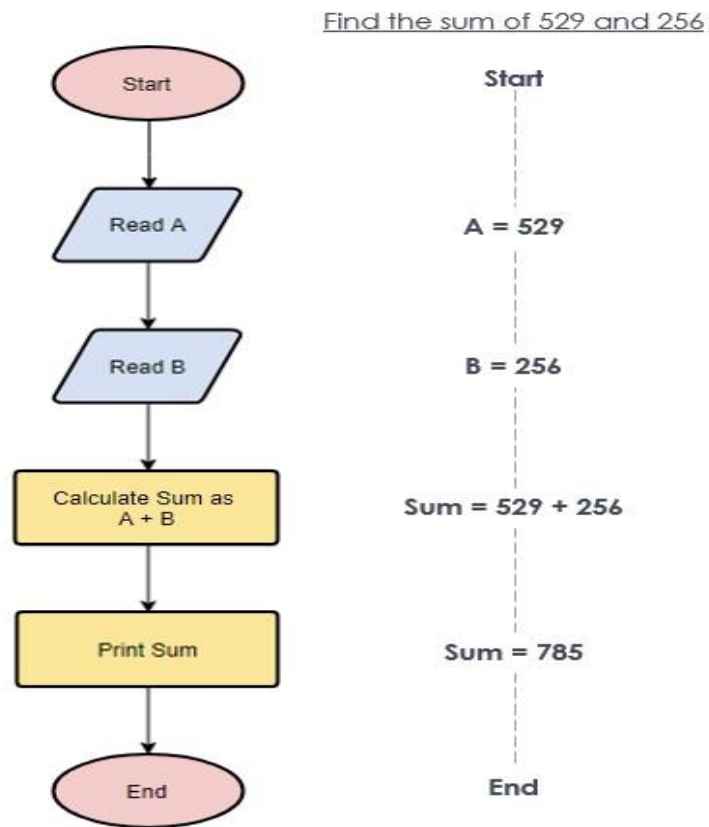
	1) Flowchart	2) Pseudocode
Start/End		Begin/End (not necessary)
Input/Output		Read/Print : read x,y
Process		[An equation] : $x=y+6$ Comput: compute x as $y+6$
Decision		If/ if-else / While
Predefined process		[function_name()]: average(x,y)
Line		Not Applicable
Connector		Not Applicable

Table 3: Pseudocode & Flow Chart

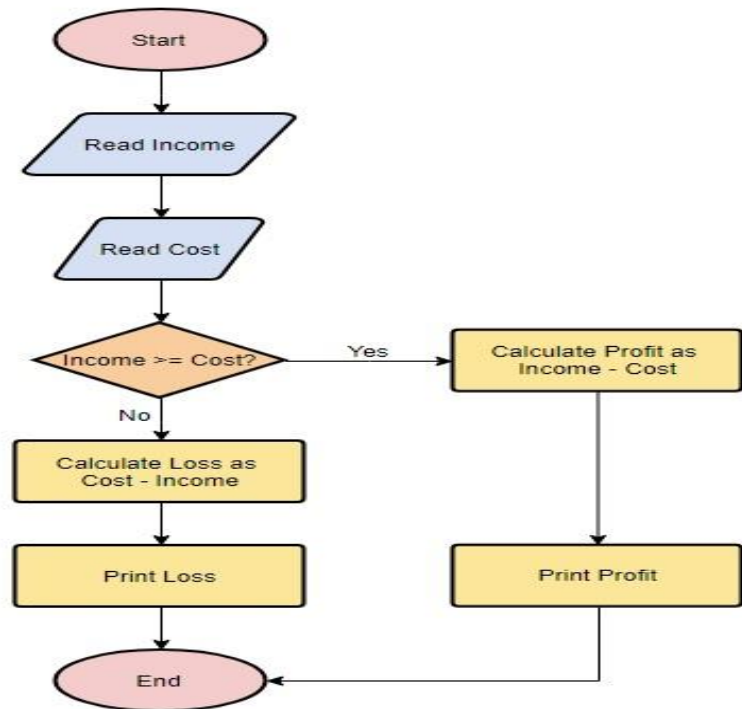
Example 1: : The flowchart example below shows a simple summation process



Figure 21: Flow Chart (1)



Example 2: The flowchart example below shows how profit and loss can be calculated



Find the profit/loss when
income = 1,000, cost = 800

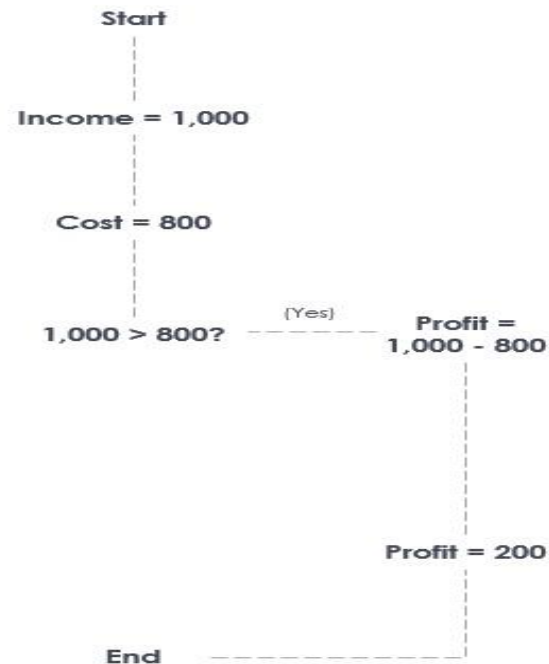


Figure 22: Flow Chart (2)

Microsoft ViSiO

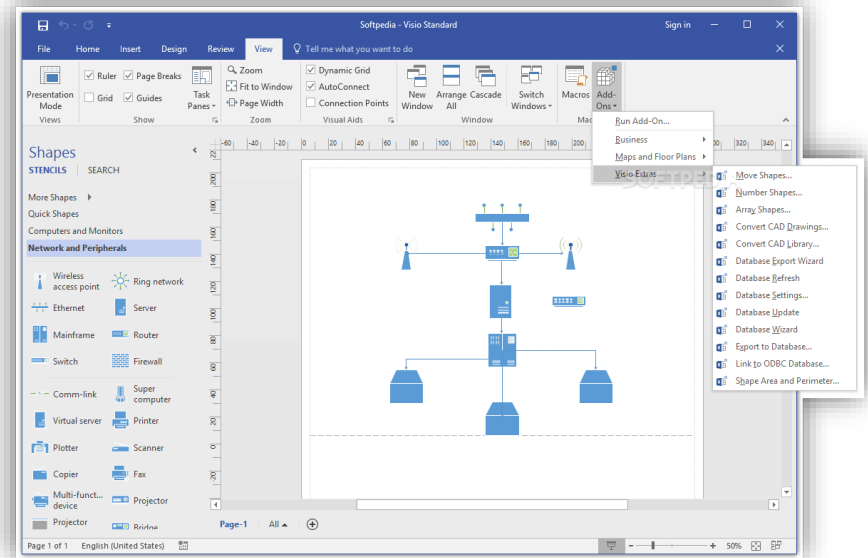
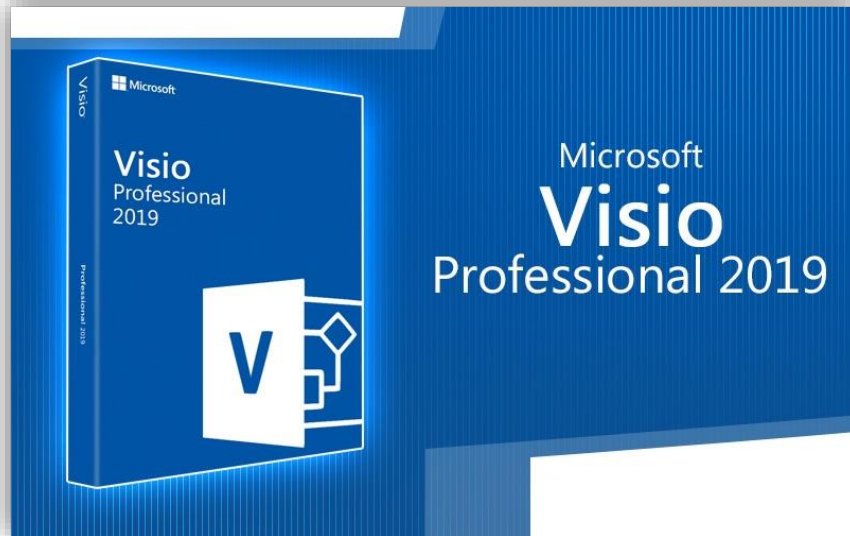


Figure 23: Microsoft Visio



Q: Draw a Flow Chart for the following :

✧ Conditions:

C1: Advance payment made

C2: Purchase amount = Rs 10,000/-

C3: Regular Customer

✧ Actions:

A1: Give 5% discount

A2: Give no discount



Q: Draw a Flow Chart for the following :

✧ Rules:

R1: Yes Advance payment made = Give 5% discount

R2: No Advanced payment made + Yes Purchase amount = Rs 10,000/- + Yes Regular Customer = Give 5% discount

R3: No Advanced payment made + Yes Purchase amount = Rs 10,000/- + No Regular Customer = Give no discount

R4: No Advanced payment made + No Purchase amount = Rs 10,000/- = Give no discount

Ru1 +R2 = Give 5% discount R3+R4 = Give no discount

Assignment 2 ☺



Guide for selecting appropriate tools



- ✧ Use **DFD** at high or low level analysis for providing good system documentations.
- ✧ Use **data dictionary** to simplify the structure for meeting the data requirement of the system.
- ✧ Use **pseudocode** English if there are many loops and actions are complex.
- ✧ Use **decision tables** when there are a large number of conditions to check and logic is complex.
- ✧ Use **decision trees** when sequencing of conditions is important and if there are few conditions to be tested.
- ✧ Use **flow chart** to clarify complex processes.



Thanks