Algoritma & Pemrograman Saintifik

Gatot F. Hertono, Ph.D

Departemen Matematika

SCMA601401

Basic Idea behind a Computer Program

- We need to determine what's relevant in the problem statement and what we can ignore.
- The goal of designing a program is to create a mechanism that consumes and produces data.
- We need to understand what the program consumes, what it produces, and how it relates inputs to outputs.
- Once we have a program, we must check whether it actually performs the intended computation. This might reveal syntax errors, run-time problems, or even logical errors.

Computer Program Strutures

A computer program may consist of a sequence of instructions that control the flow of the program. It may contain:

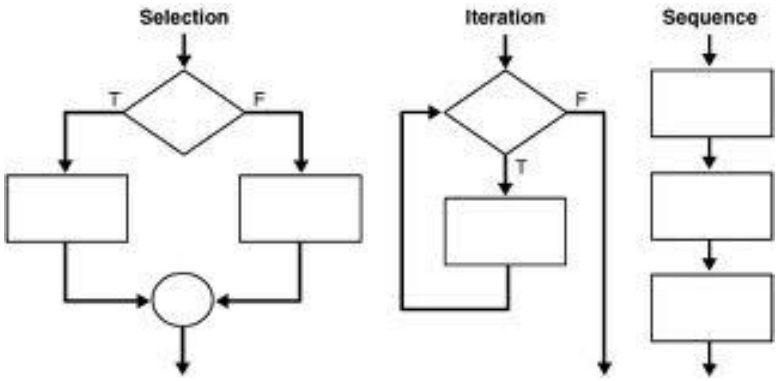
Simple statements: contain a single statement/command or (arithmetic) assignments

Conditional (selection) statements could contain :

- Single selection *(IF (...) THEN ... ELSE ...)*
- Multiple selection (SELECT CASE (...) DO ...)

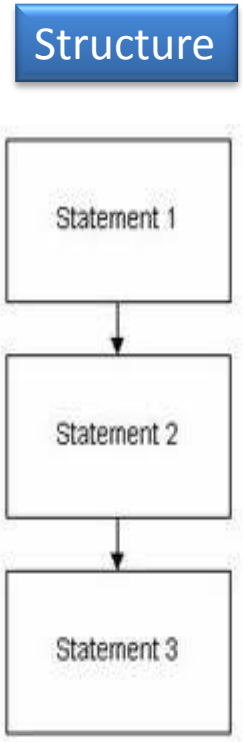
Repetition/Iteration (loop) statements could be:

- Fixed loop (*FOR var* = ... *TO* ... *DO* ...)
- Conditional loop
  - WHILE (...) DO ...
  - REPEAT ... UNTIL (...)

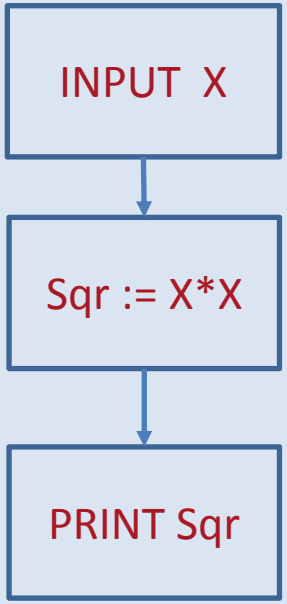


Simple Statements

Example:



Flowchart



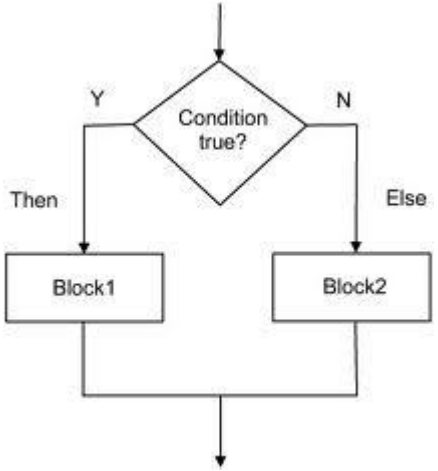
Pseudocode

```
INPUT:   X
OUTPUT:  Sqr
PROCESS:
   input   X
   Sqr  :=  X*X
   print   Sqr
```

Conditional Statements

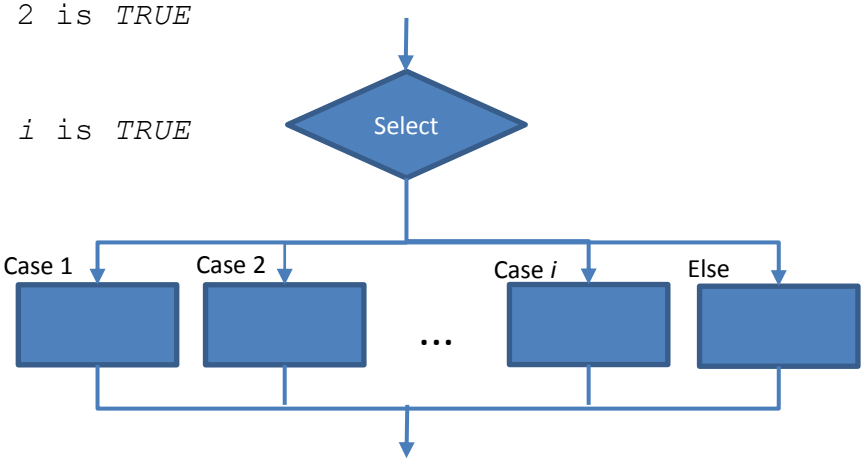
- Case 1: single selection

```
IF: (test condition) THEN
   Statement(s) to be executed if test condition is TRUE
ELSE:
   Statement(s) to be executed if test condition is FALSE
```



- Case 2: multiple selection

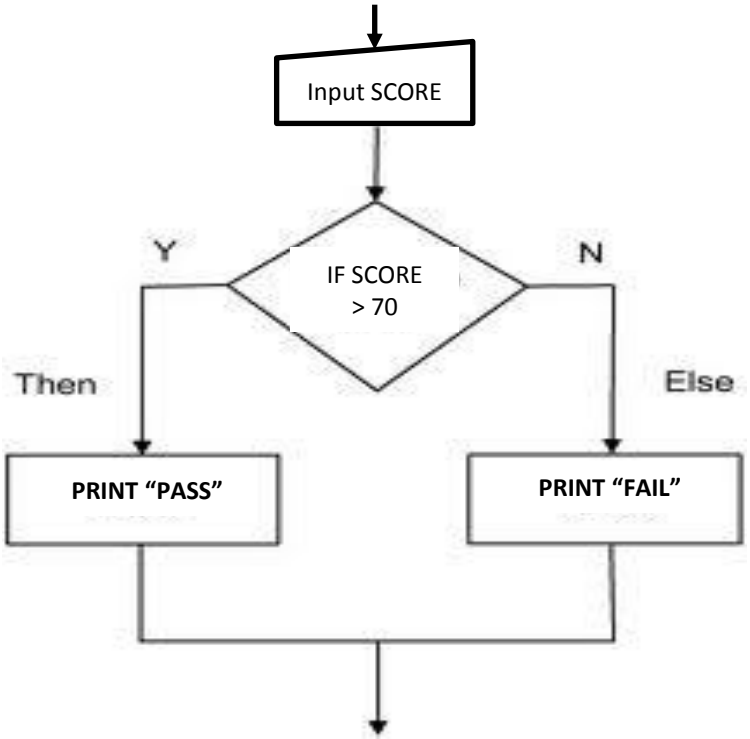
```
SELECT
   CASE(test condition 1) DO
      Statement(s) to be executed if test condition 1 is TRUE
   CASE(test condition 2) DO
      Statement(s) to be executed if test condition 2 is TRUE
   ...
   CASE(test condition i) DO
      Statement(s) to be executed if test condition i is TRUE
   ...
   ELSE
      Statement(s) to be executed if otherwise
```



Conditional Statements

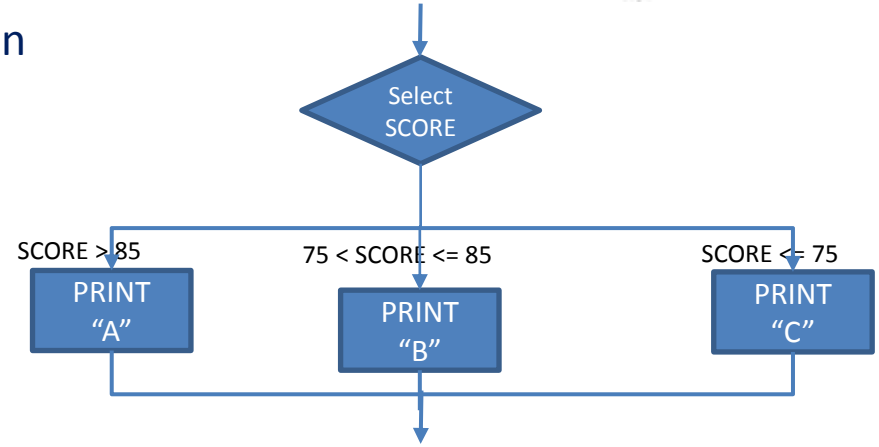
- Case 1: single selection

```
INPUT: SCORE
OUTPUT: "PASS"/"FAIL"
PROCESS:
INPUT SCORE
IF SCORE > 70 THEN
   PRINT "PASS"
ELSE:
   PRINT "FAIL"
```

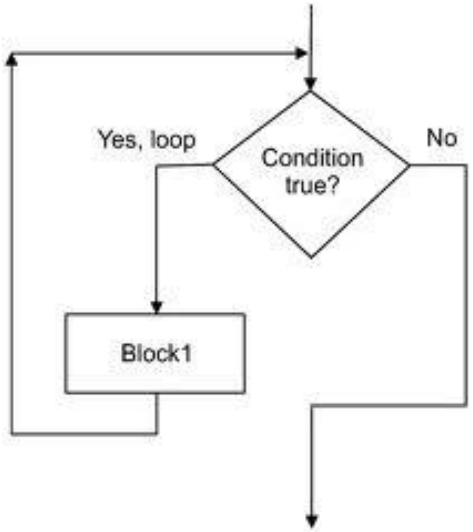


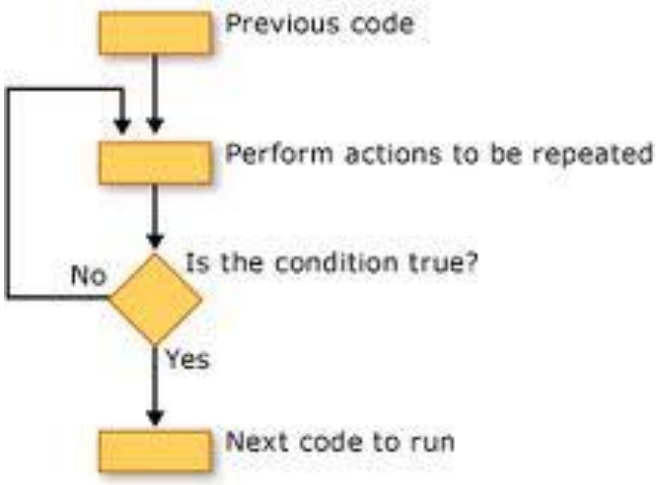
- Case 2: multiple selection

```
...
SELECT
   CASE(SCORE > 85) DO
      PRINT "A"
   CASE(SCORE > 75) DO
      PRINT "B"
   ELSE
      PRINT "C"
```



Repetition Statemens





FOR *variable* = ... **TO** ... **DO**
statements

(exact number of loops)

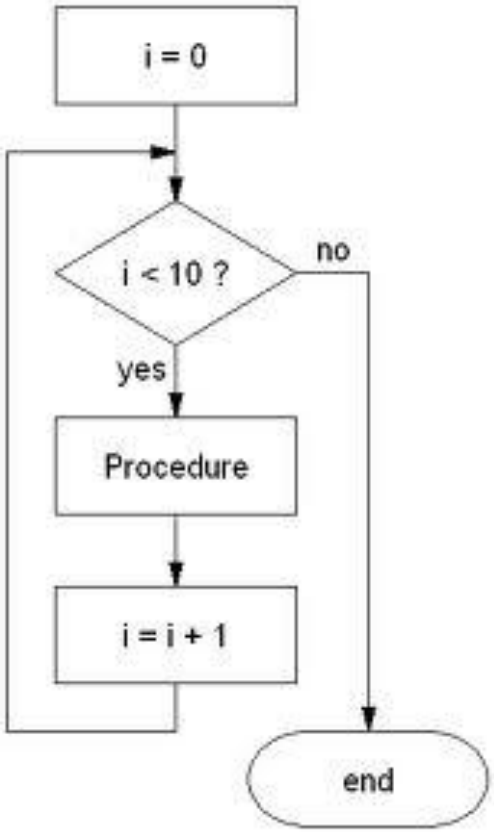
WHILE (test condition) **DO**
Statement(s) to be executed if test condition is *TRUE*

(probably no loops executed)

REPEAT
Statement(s) to be executed if test condition is *FALSE*
UNTIL (test condition True)

(at least one loops executed)

Repetition Statements




```
FOR i = 0 TO 9 DO
      Procedure
```

```
i:=0
WHILE (i < 10) DO
    Procedure
    i := i + 1
```

```
i:=0
REPEAT
    Procedure
    i := i + 1
UNTIL (i >= 10)
```

Exercise: Use a flowchart to represent the above algorithm with the REPEAT... UNTIL structure

Exercises

1. Use a single selection structure to represent multiple selections (in flowchart and pseudocode).
2. Use a single selection structure to represent fixed loop (in flowchart)
3. Describe a flowchart to compute: $\Pi_{i=1}^{n} i$