

MMJ 1113

Computational Methods for Engineers

Programming and Software

Introduction

- Objective is how to use the computer as a tool to obtain numerical solutions to a given engineering model. There are two ways in using computers:
 - ◆ Use available software, or
 - ◆ write computer programs to extend the capabilities of available software, such as Excel and Matlab
- Engineers should not be “tool-limited”, it is important that they should be able to do both!
- Computer programs are set of instructions that direct the computer to perform a certain task.

Introduction

- To be able to perform engineering-oriented numerical calculations, you should be familiar with the following programming topics:
 - ◆ Simple information representation (constants, variables, and type declaration)
 - ◆ Advanced information representation (data structure, arrays, and records)
 - ◆ Mathematical formulas (assignment, priority rules, and intrinsic functions)
 - ◆ Input/Output
 - ◆ Logical representation (sequence, selection, and repetition)
 - ◆ Modular programming (functions and subroutines)

Structured Programming

- *Structured programming* is a set of rules that prescribe good style habits for programmer.
 - ◆ An organized, well structured code
 - ◆ Easily sharable
 - ◆ Easy to debug and test
 - ◆ Requires shorter time to develop, test & update
- The key idea is that any numerical algorithm can be composed of using the three fundamental structures:
 - ◆ Sequence, selection & repetition

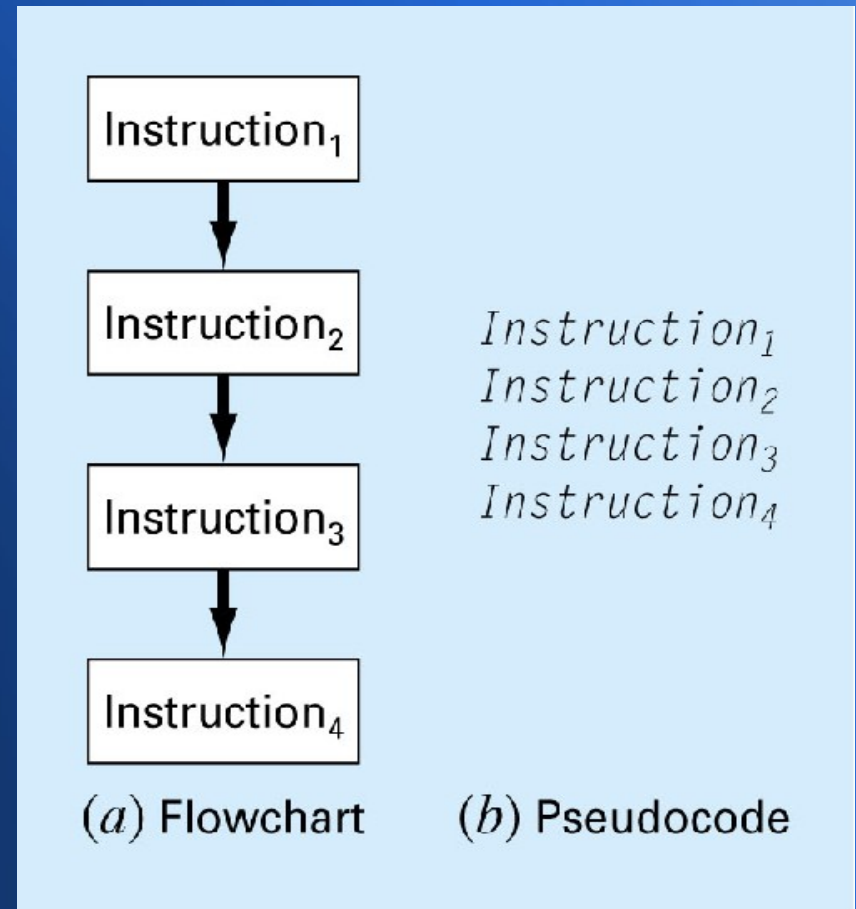
Structured Programming

SYMBOL	NAME	FUNCTION
	Terminal	Represents the beginning or end of a program.
	Flowlines	Represents the flow of logic. The humps on the horizontal arrow indicate that it passes over and does not connect with the vertical flowlines.
	Process	Represents calculations or data manipulations.
	Input/output	Represents inputs or outputs of data and information.
	Decision	Represents a comparison, question, or decision that determines alternative paths to be followed.
	Junction	Represents the confluence of flowlines.
	Off-page connector	Represents a break that is continued on another page.
	Count-controlled loop	Used for loops which repeat a prespecified number of iterations.

Structured Programming

- **Sequence**

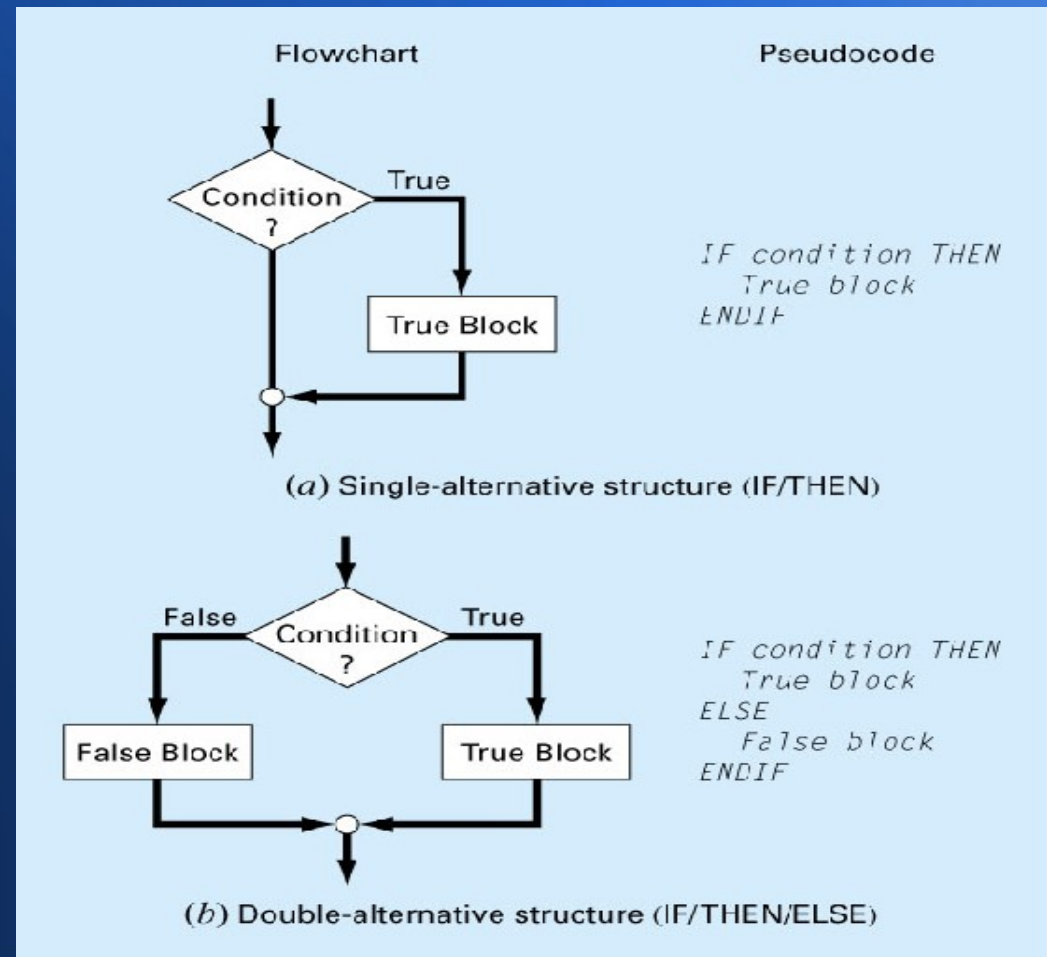
Computer code is implemented one instruction at a time, unless instructed otherwise. The structure can be expressed as a flowchart or pseudocode



Structured Programming

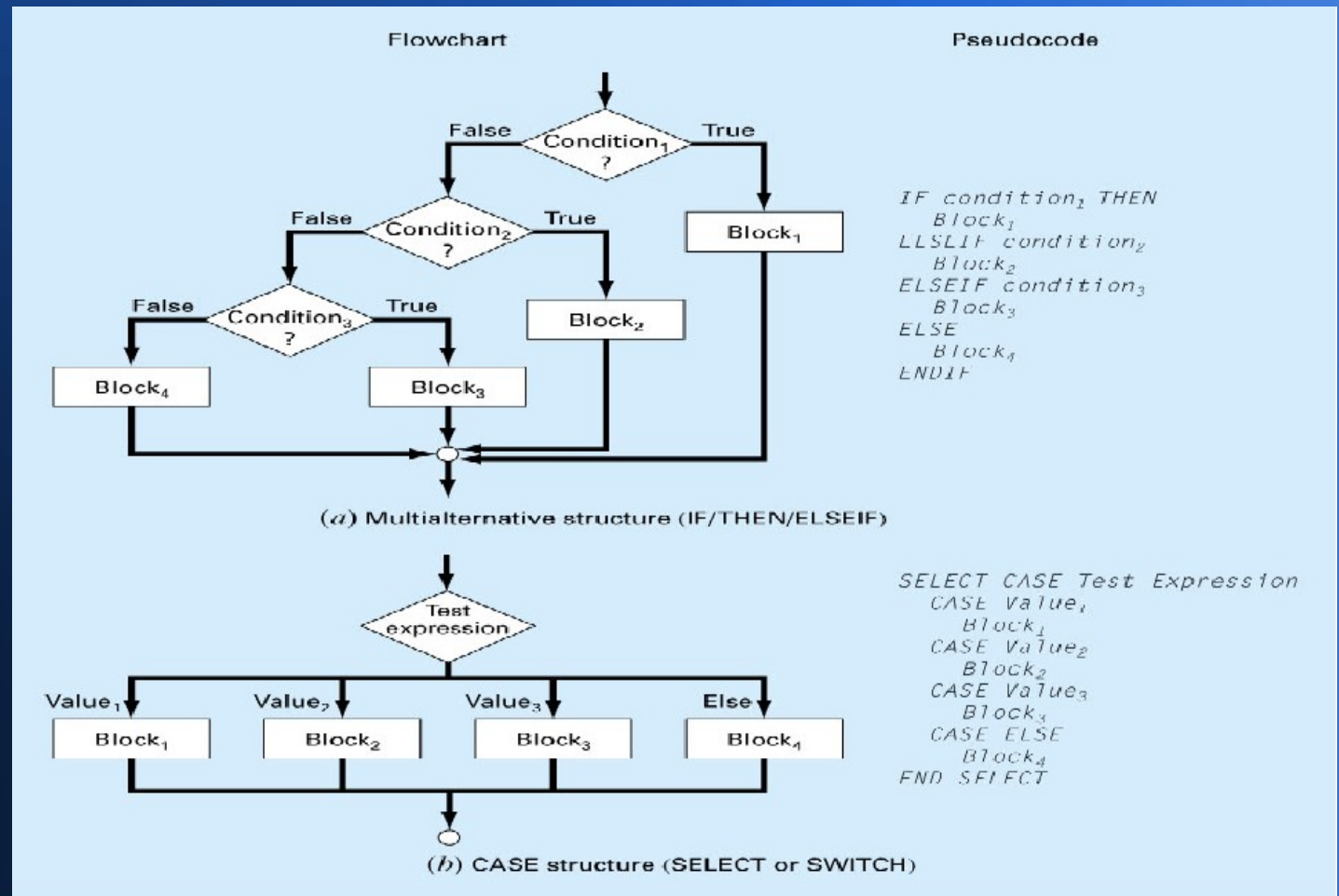
- **Selection**

Splits the program's flow into branches based on outcome of a logical condition



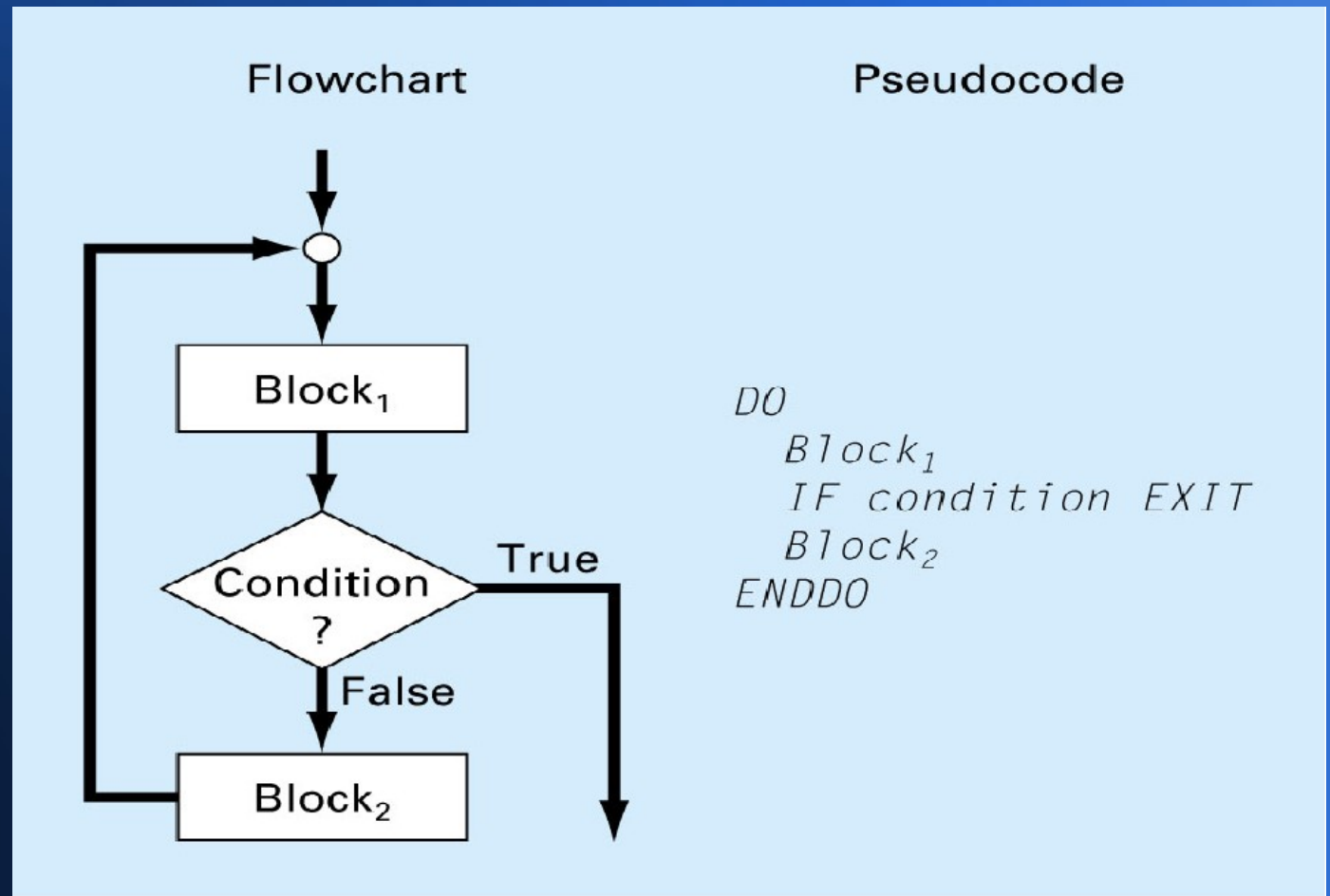
Structured Programming

- Repetition
(E.g. 1)
Implement instructions repeatedly



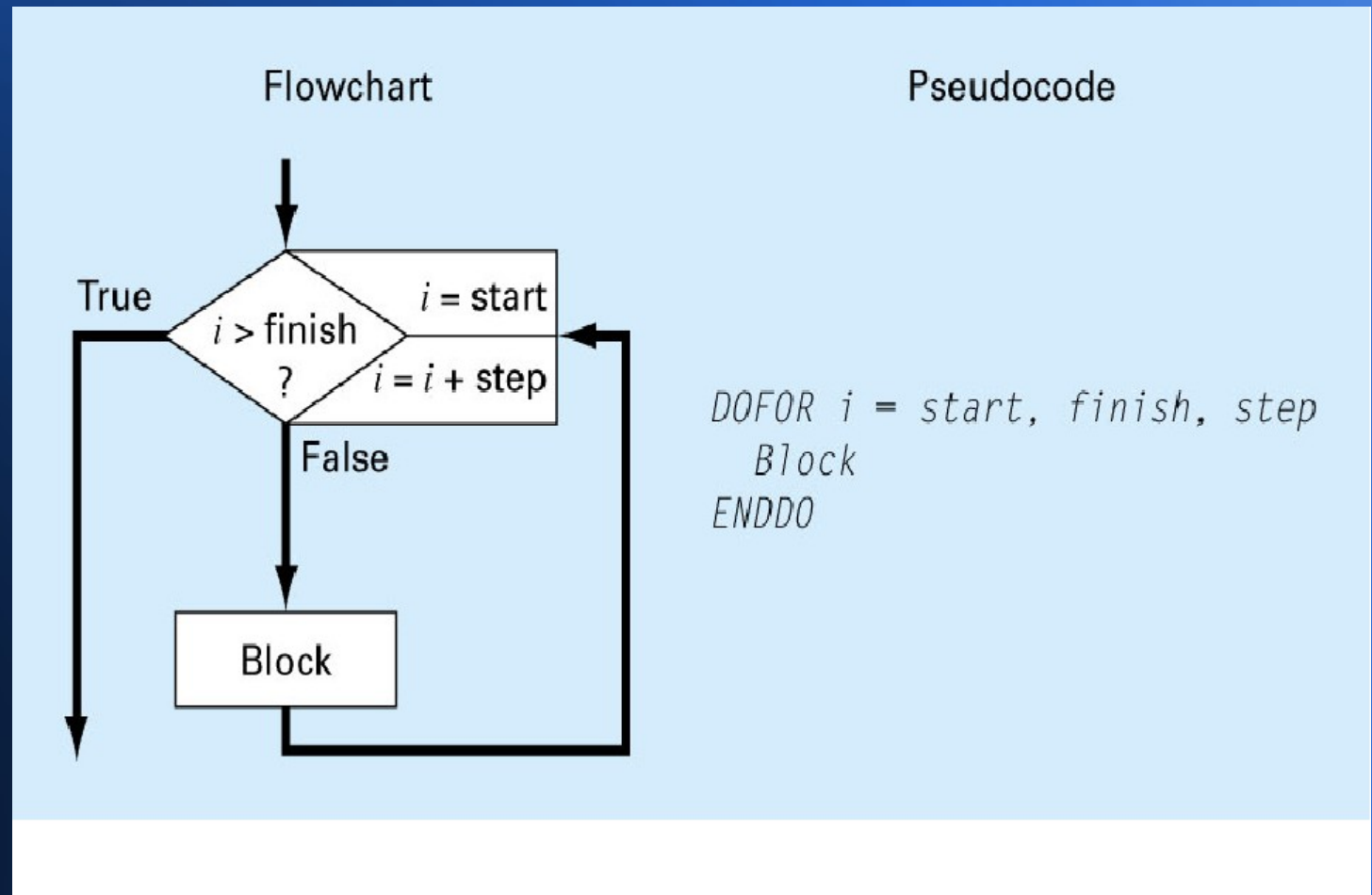
Structured Programming

- **Repetition**
(E.g. 2)
Implement
instructions
repeatedly



Structured Programming

- **Repetition**
(E.g. 3)
Implement instructions repeatedly



Modular Programming

- The computer programs can be divided into subprograms, or **modules**, that can be developed and tested separately
- Modules should be as independent and self contained as possible
- Advantages to modular design are:
 - ◆ Easier to understand the underlying logic of smaller modules
 - ◆ Easier to debug and test
 - ◆ Facilitate program maintenance & modification
 - ◆ Programmer can maintain own library of modules for later use

Modular Programming

```
FUNCTION Euler(dt, ti, tf, yi)
  t = ti
  y = yi
  h = dt
  DO
    IF t + dt > tf THEN
      h = tf - t
    ENDIF
    dydt = dy(t, y)
    y = y + dydt * h
    t = t + h
    IF t ≥ tf EXIT
  ENDDO
  Euler = y
END
```

• Programming Languages

- C/C++
- Matlab (Octave/Scilab)
- Fortran 77/9X/2003
- Python