### Introduction to Programming

**Bertrand Meyer**

#### Lecture 8:
**Control structures (1)**

Last revised 20 November 2004

---

### The notion of algorithm

**General definition:**

An algorithm is the specification of a process to be carried out by a computer.

---

### Not quite an algorithm

#### PREPARAZIONE E TEMPI DI COTTURA

* Zubereitung - Preparation *

Verso la vostra zuppa unguentata, 1 litro abbondante d’acqua fredda con 2 cucchi di sale e coccio secondo i tempi indicati. Tegliettare Gemjou in un litro caldo. Rosolare 2 chilini di olio e Sal noci schiacciate. Veneri le legumi soffritti in 1 litro d’acqua fredda, aggiungere due cialde a soupe d’huile et du sel.

---

### 5 properties of an algorithm

1. Defines data to which process will be applied
2. Every elementary step taken from a set of well-specified actions
3. Describes ordering(s) of execution of these steps
4. 2 and 3 based on precisely defined conventions, suitable for execution by an automatic computer
5. For any data, guaranteed to terminate after finite number of steps

---

### Algorithm vs program

- "Algorithm" usually considered a more abstract notion, independent of platform, programming language etc.
- In practice, the distinction tends to fade:
  - Algorithms need a precise notation
  - Programming languages becoming more abstract
- However:
  - In programs, data (objects) are just as important as algorithms
  - A program typically contains many algorithms and object structures

---

### What makes up an algorithm

- **Basic steps:**
  - Feature call $x, f(a)$
  - Assignment (yet to be studied)
  - ...

- **Sequencing of these basic steps:**

  **CONTROL STRUCTURES**
Control structures

Definition: program construct that describes the scheduling of basic actions

Three fundamental control structures:
- Sequence
- Loop
- Conditional

"Control structures of Structured Programming"

Control structures as problem-solving techniques

- Sequence: "To achieve C from A, first achieve an intermediate goal B from A, then achieve C from B"
- Loop: solve the problem on successive approximations of its input set
- Conditional: solve the problem separately on two or more subsets of its input set

The sequence (or Compound)

instruction₁
instruction₂
...
instructionₙ

Semicolon as optional separator

instruction₁;
instruction₂;
...
instructionₙ

Correctness of a Compound

Precondition of instruction₁ must hold initially
Postcondition of each instruction, must imply precondition of each instruction₂
Final effect is postcondition of instructionₙ

Conditional instruction

if Condition -- Boolean_expression
then Instructions -- Compound
else Other_instructions -- Compound
end
The conditional as problem-solving technique

PROBLEM SPACE

Region 2
Region 1
Use technique 1
Use technique 2

Basic form

if Condition then
  Instructions
else
  Other_instructions
end

A variant of the conditional

if Condition then
  Instructions
end

(as if "else ")

Nesting

if Condition then
  Instructions_1
else
  if Condition_1 then
    Instructions_2
  else
    if Condition_2 then
      Instructions_3
    else
      Instructions_4
  end
end
Nesting

```plaintext
if Condition, then
  Instructions1
else
  if Condition, then
    Instructions2
  else
    if Condition, then
      Instructions3
    else
      Instructions4
    end
  end
end
```

Nested structure

Comb-like structure

```plaintext
if Condition, then
  Instructions1
else if Condition, then
  Instructions2
else if Condition, then
  Instructions3
else
  ...
else
  Instructions5
end
```

Comb-like conditional

Comb-like structure

End of lecture 8