Representation of Algorithms

- Natural languages (English, Spanish, Chinese, …)
- Formal programming languages (Java, Pascal, C++, …)
- Pseudocode
  - "a special set of English language constructs modeled to look like the statements available in most programming languages"
  - compromise between natural & formal languages

Sequential Operations

Input Operation

- Example:
  Get a value for $r$, the radius of the circle

  \[
  \text{Get a value for } r \quad 25, 4
  \]

- General form:
  Get values for "variable", "variable", …
- "variable" means "any variable" (e.g., $r$)

Output Operation

- Example:
  Print the value of Area

  \[
  \text{Print the value of Area} \quad 26,4
  \]

- General form:
  Print the values of "variable", "variable", …

Output Operation (2)

- Example:
  Print the message 'Sorry, no answer'

  \[
  \text{Print the message 'Sorry, no answer'} \quad \text{Sorry, no answer, 4}
  \]

- General form:
  Print the message 'message'
Computational Operation

- Example: Set the value of Area to \( \pi r^2 \)
  \[
  \pi = 3.14... \quad \text{compute} \quad 3.14... \times (25)^2 \quad \text{Area} \quad = 196 \quad \text{g}
  \]
- General form: Set the value of “variable” to “arithmetic expression”

Computation is Physically Instantiated

\[
\pi \times (2 \text{ cm}/2)^2 \times 3 \text{ cm} \times 13.55 \text{ g/cm}^3 = 127.7 \text{ g}
\]

Computational Processes in General

- Computation takes place in a medium
  - in which patterns can be represented
  - by means of which they can be manipulated
- Example computational media:
  - electrical signals & charges
  - light intensity, fluid pressure, molecular structure, quantum state, gear position (old), abacus beads, paper, blackboard, etc.

Input Processes in General

- An input process translates an external physical situation into the computer’s internal representational medium
- Example physical situations:
  - position of mouse or joystick
  - position of key (depressed or not)
  - signal from light sensor
  - etc.

Output Processes in General

- An output process translates the computer’s internal medium of representation into an external physical effect
- Example physical effects:
  - characters on display screen or printer
  - graphics images
  - position of robot arm
  - etc.

Conditional Operation

- if true
- \( \text{true/false condition} \)
- do one set of operations
- if false
- \( \text{do another set of operations} \)
- continue
Conditional Statement

**General:**
If “true/false condition” is true then
"first set of algorithmic operations"
Else (or Otherwise)
"second set of algorithmic operations"
(rest of algorithm)

**Example:**
If \((c_i \geq 10)\) then
Set the value of \(c_i\) to \((c_i - 10)\)
Set the value of \(carry\) to 1
Else
Set the value of \(carry\) to 0

---

One-Branch Conditional

If “true/false condition” is true then
“first set of algorithmic operations”
(rest of algorithm)

---

Computer-Inspired Conditionals in Legal Drafting

Subsection (a). IF AND ONLY IF
(1) (A) A person has threatened or attempted suicide or to inflict serious bodily harm on himself, OR
(B) The person has threatened or attempted homicide or other violent behavior, OR
(C) The person has placed others in reasonable fear of violent behavior and serious physical harm to them, OR
(D) The person is unable to avoid severe impairment or injury from specific risks, AND
(2) There is a substantial likelihood that such harm will occur,
THEN
(3) The person poses a “substantial likelihood of serious harm” for purposes of subsection (b).

---

Basic Loop (Trailing Decision)

Repeat steps \(i\) to \(j\) until “a true/false condition” becomes true

---

Example Loop

1. Set the value of \(count\) to 1
2. Repeat steps 3 to 5 until \((count > 100)\)
3. Set \(square\) to \((count \times count)\)
4. Print the values of \(count\) and \(square\)
5. Add 1 to \(count\)
Alternate Notation

Repeat until “a true/false condition” becomes true
“a set of algorithmic operations”
End of loop

Leading Decision Loop

While “a true/false condition” remains true do
“a set of algorithmic operations”
End of loop