Pseudo Language Cheat Sheet

The computer languages such as Java, C, Python are somewhat complicated to learn at first. Instead of trying to learn a “real” language, we will now introduce a Pseudo language in order to understand algorithmic thinking first.

Flow Controls

SEQUENCE is a linear progression where one task is performed sequentially after another.

WHILE is a loop (repetition) with a simple conditional test at its beginning.

IF-THEN-ELSE is a decision (selection) in which a choice is made between two alternative courses of action.

Category	Instruction	Syntactical Form	Example
INPUT/OUTPUT (I/O)	**READ**	**READ** <variable>	**READ** height
	DISPLAY	**DISPLAY** <variable/const>	**DISPLAY** height
ARITHMETIC	**ADD** **SUBTRACT** **MULTIPLY** **DIVIDE**	**ADD** <var/const> to y<var/const> **SUBTRACT** <var/const> from <var/const> **MULTIPLY** <var/const> by <var/const> **DIVIDE** <var/const> by <var/const>	**ADD** x to 4: x ➔ x +4 **SUBTRACT** 4 from x: x ➔ x-4 **MULTIPLY** x by 4: x ➔ 4*x **DIVIDE** x by 4: x ➔ x/4
INITIALIZATION	**SET**	**SET** <variable> to <var/const>	**SET** x to 5: x➔ 5
COMPUTATION	**COMPUTE**	**COMPUTE** <expression>	**COMPUTE** area as height*width

BRANCHING (Binary)	**IF-ELSE**	**IF** <Boolean expression>
		<Statement if Boolean expression is true>
		ELSE
		<Statement if Boolean expression is false>
		ENDIF
			IF temperature is above 90
			DISPLAY the message it is hot today
			ELSE
			DISPLAY the message today is comfortable
			ENDIF

BRANCHING (multi)	**CASE**	CASE **<expression>** **Case <value>: <statements>** **Case <value>: <statements>** **….** **Default: <Statements>** **ENDCASE**	**CASE x** **0 : DISPLAY “zero”** **1: DISPLAY “one”** **Default: DISPLAY “not one or zero”** **ENDCASE**
REPETITION	**WHILE**	**WHILE** <Boolean expression> <Statement if Boolean expression is true> **ENDWHILE**	**WHILE** (n > 0) ADD n to sum SUBTRACT 1 from n **ENDWHILE**
	DO.. WHILE	**DO** <statement/s> **WHILE** condition	**DO** SUBTRACT 1 from n **WHILE** (n > 0)
	REPEAT-UNTIL	**REPEAT** <statement/s> **UNTIL** condition	**SET** n to 10 **REPEAT** ADD 1 to n **UNTIL** (n > 100)
	FOR	**FOR** <iteration bounds> <statement/s> **ENDFOR**	**FOR** each day of the week DISPLAY day of the week **ENDFOR**
NESTED	**NESTED-FOR**	**FOR** <iteration bounds> **FOR** iteration bounds <statement/s> **ENDFOR** **ENDFOR**	**FOR** each day of the month **FOR** each day of the week **DISPLAY** day of the month + day of week **ENDFOR** **ENDFOR**

	NESTED-FOR-IF	**FOR** <iteration bounds>
		IF <Boolean expression>
		<statement/s>
		ELSE
		<statements>
		ENDIF
		ENDFOR
			FOR day of the month
			IF day is even
			DISPLAY “good day”
			ELSE
			DISPLAY “odd day”
			ENDIF
			ENDFOR

DEFINITIONS

<variable> or **<var>**: An entity that may change its value (eg. X)
<constant> or **<const>:** A value that does not change (eg. 4)
<value> : A value of an expression
<Boolean expression>: An expression that is TRUE or FALSE (eg. N > 3)
<Statement/s>: One or more statements in a program. (eg: x = x + y)
<Iteration bounds>: a set of values where program is iterating on. (eg. **FOR** days of the week)