Control structures

- Control structures are used to control the logical flow of instructions during the execution of a program

- The following are the basic types of control structures:

  Sequential - by default the instructions are executed one after another

  Selection - based on logical decisions enable the execution of 2 or more alternative branches

  Repetition - based on logical decisions or iterating through values allow the repetition of a block of code multiple times (loop)

Logical variables and expressions

- Python has a Boolean data type that is either **True** or **False**
- A variable is of type Boolean if it has assigned a value of **True** or **False**
- Selection and Repetition control structures use the value, True or False, of logical expressions to decide what instructions to execute next

Relational operators

- Relational operators are used to compare two values that should be of the same data type (both numerical or both string, …)

OPERATOR	MEANING	OPERATION
==	EQUAL	X == Y
!=	NOT EQUAL	X != Y
<	LESS THAN	X < Y
>	GREATER THAN	X > Y
<=	LESS THAN OR EQUAL TO	X <= Y
>=	GREATER THAN OR EQUAL TO	X >= Y

Relational operators

String comparison

- Strings are compared character by character until a decision can be made
- The characters are compared using their Unicode value.

Ex:

Expression	Value
"AA" < "AB"	True
"14" > "200"	False
"a" < "A"	False

Expression	Value
"AB " > "AB"	True
"2Z" > "AZ"	False
"AB" = "AB"	True

Logical Operators

- Logic operators perform logical operations on logical values (True or False)

OPERATOR	OPERATION
NOT	NEGATION
AND	CONJUNCTION
OR	DISJUNCTION

X	Y	not X	X and Y	X or Y
False	False	True	False	False
False	True	--	False	True
True	False	False	False	True
True	True	--	True	True

Computing expressions - priorities

- Expressions can be formed by:
  - **ARITHMETIC EXPRESSIONS** – The result is a numerical value
  - **STRING EXPRESSIONS** – The result is a string value
  - **RELATIONAL OPERATORS** – A pair of values (numerical, string, …) can be compared resulting in a logical value
  - **LOGICAL OPERATORS** – A logical operation can be performed on a pair of logical values (True or False) resulting in a logical value

PRIORITIES:

1º - FUNCTIONS
2º - ARITHMETIC OPERATIONS
3º - RELATIONAL OPERATIONS
4º - LOGICAL OPERATIONS

The order of calculation can be changed by the use of parenthesis. In the same circumstances operations are carried out from left to right

Ex: A > B **or** D < C **and** A * B / **sin**(D) <= 10

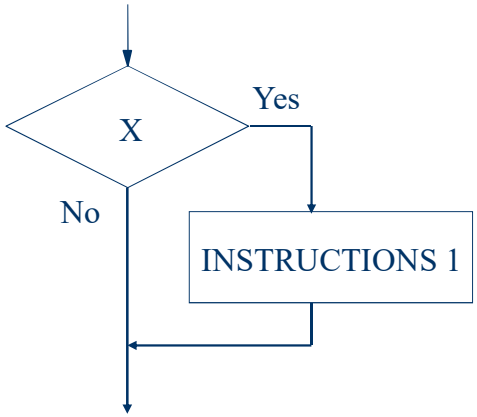
4º 5º 2º 1º 3º 6º 7º 8º

Selection structure

- Executes a block of statements only if the logical expression is **True**

FLOWCHART

X – Logical expression



PSEUDOLANGUAGE

```
IF X THEN
   INSTRUCTIONS 1
END IF
```

PYTHON

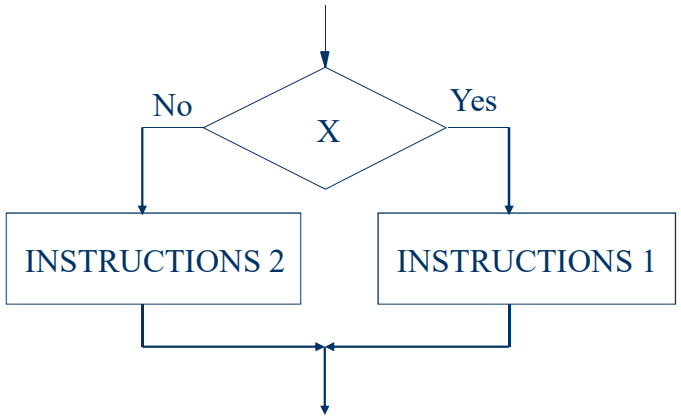
```
if X:
    INSTRUCTIONS 1
```

Selection structure

- Executes a block of statements (INSTRUCTIONS 1) if the logical expression is **True** and another block of statements (INSTRUCTIONS 2) if the logical expression is **False**

FLOWCHART

X – Logical expression



PSEUDOLANGUAGE

```
IF X THEN
   INSTRUCTIONS 1
ELSE
   INSTRUCTIONS 2
END IF
```

PYTHON

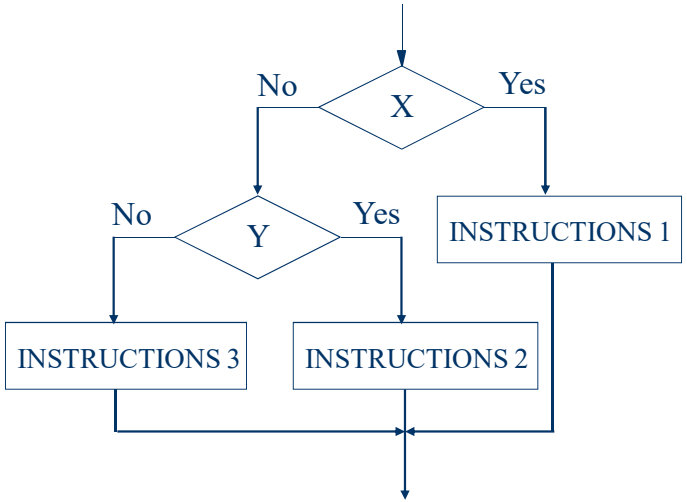
```
if X:
   INSTRUCTIONS 1
else:
   INSTRUCTIONS 2
```

Selection structure

- If the condition (**X**) is **True**, executes the block of statements (INSTRUCTIONS 1). If the condition (**X**) is False it checks the condition of the next **elif** block and so on. If the **if** condition and all the **elif** conditions are **False** the block **else** is executed.

FLOWCHART

X, Y – Logical expressions



PSEUDOLANGUAGE

```
IF X THE
   INSTRUCTIONS 1
ELSEIF Y THEN
   INSTRUCTIONS 2
ELSE
   INSTRUCTIONS 3
END IF
```

PYTHON

```
if X:
    INSTRUCTIONS 1
elif Y:
    INSTRUCTIONS 2
else:
    INSTRUCTIONS 3
```