

Component 1: Algorithms and Constructs

Term	Definition
Algorithm	In programming, an algorithm is a set of instructions that can be used to solve a given problem.
Pseudo code	A notation resembling a simplified programming language that is used in designing computer programs.

When writing an algorithm, the instructions must be clear and in the correct order to produce the required solution to the given problem.

Algorithms are written in pseudo code. The pseudo code will be presented using the following conventions:

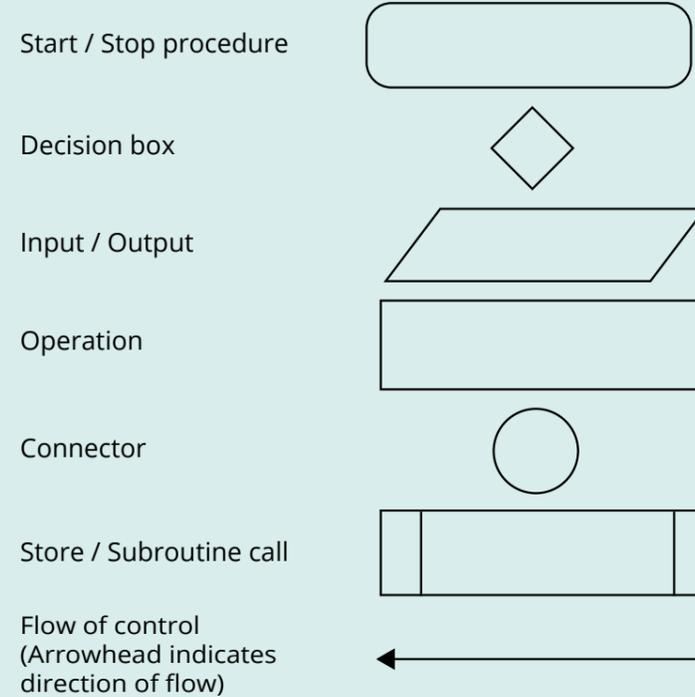
Construct	Example usage
Declare subroutines	Declare CapitalLetterOfName End Subroutine
Call a subroutine	call SubroutineNeeded
Declare and use arrays	myarray[99]
Literal outputs	output "Please enter a number"
Variable names	myvariable
Define variable data type	myvariable is integer
Data types	integer, character, string, boolean
Assignment	set counter = 0
Selection	if . . . else . . . end if
Indent at least single space after if or do or repeat etc.	if counter = 1 output counter end if
Annotation	(Some annotation goes here)
Comments (for Java only)	/** Comments for Java **/
Repetition	for i . . . next i repeat . . . until do . . . loop do . . . while while . . . repeat

Logical operators AND NOT XOR will be upper case

Logical TRUE and FALSE will be upper case

Algorithms can also be presented using flowcharts.

Sequence, selection and iteration

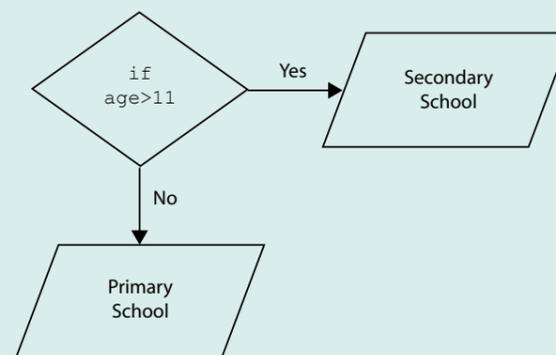


Algorithms consist of series of instructions in a specific order. This is the order or **sequence** in which the instructions must be carried out for the algorithm to work.

A **selection** instruction is one where a decision must be made. An instruction in an algorithm will give different options for the next instruction.

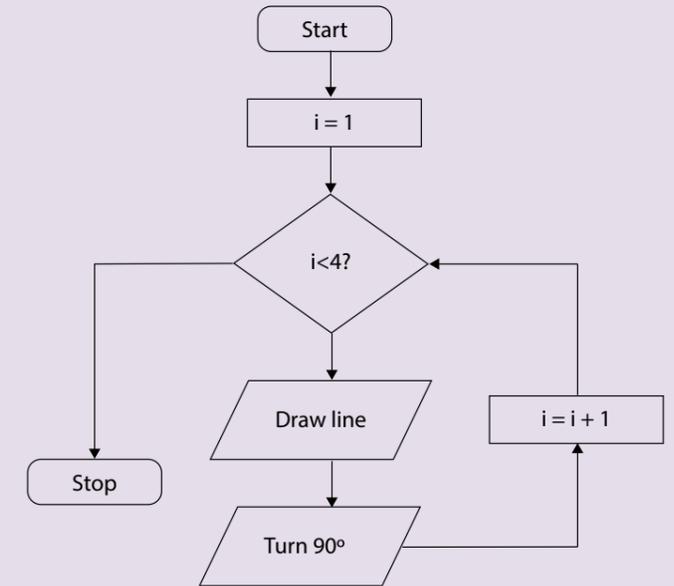
If a set of steps to be carried out more than once or many times. This is called **iteration** and often referred to as a loop in the program.

```
if age > 11 then
    print "Secondary school"
else
    print "Primary School"
end if
```



If a program must repeat a set of instructions four times, we can use a 'for i...next' loop.

Using count and rogue values with loops



All loops must eventually be terminated.

If a loop must be repeated a known number of times, a **count** can be used. When the count reaches the required number, the loop will terminate.

```
count = 0
repeat
    input data
    count = count + 1
until count = 10
```

A **rogue** value is a value that falls outside the range of possible values for the data being processed that will cause the loop to terminate.

```
total = 0
repeat
    input age
    if age > 0 then
        total = total + age
    end if
until age = -1
```