Pseudocode and Flowchart materials selected from BWagner.com

Part 1: Flowcharts

Part 2: Pseudocode

Part 1: Flowcharts

Flowchart

What You Will Learn

- 4.2.4 Analyse an algorithm presented as a flow chart.

Flowchart

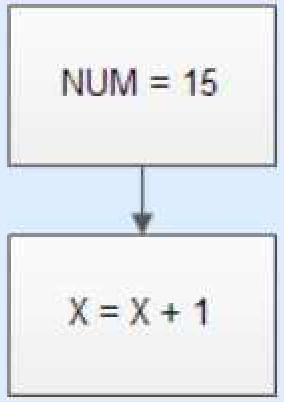
Flowchart - a graphical representation of an algorithm used to solve a programming problem.

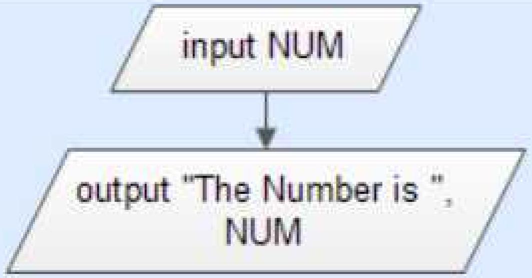
The chart below shows the different symbols used in a flowchart to describe the flow of an algorithm.

When will you use a flow chart ?

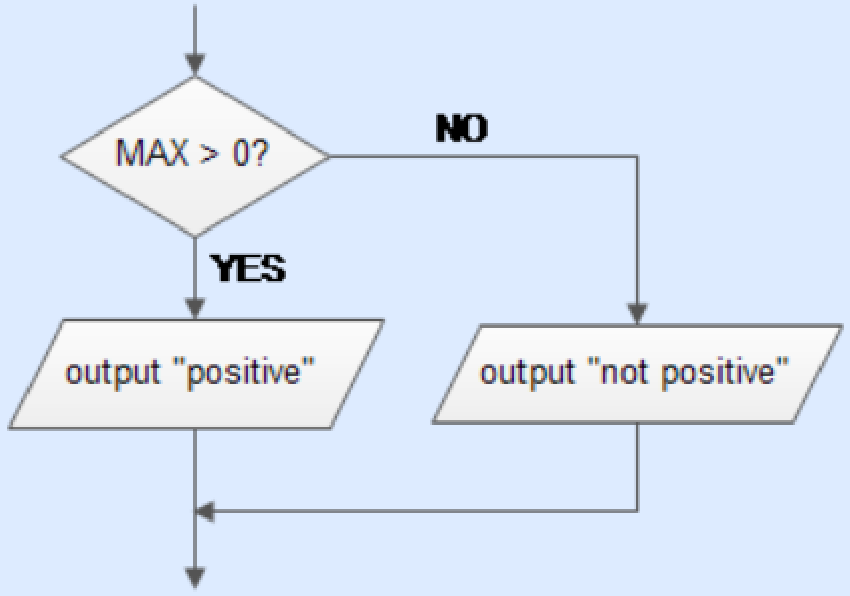
- when developing an algorithm
- when trying to break-down a problem
- to show the steps required to solve a problem like on your Internal Assessment
- You must be able to read and interpret flow charts on exams
- You will come across them as solutions to problems that are non-programming language specific.

Operation	Flowchart Example	Pseudocode Example

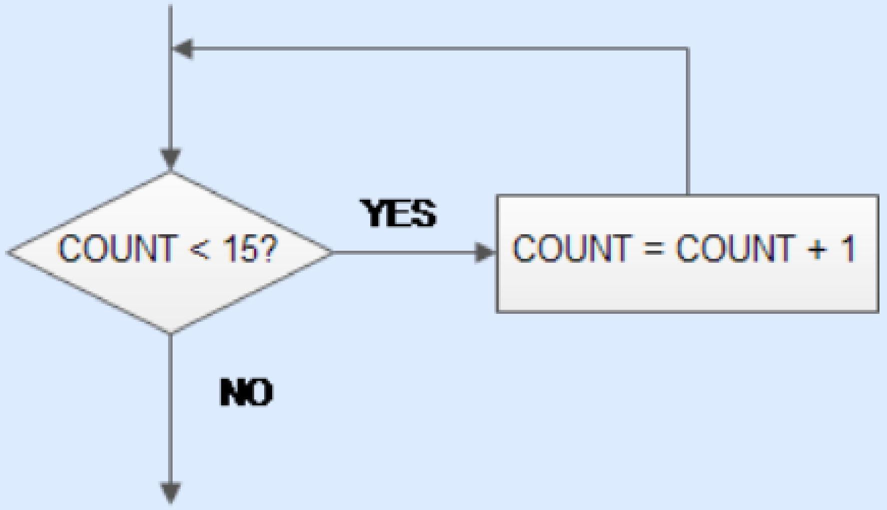
Sequential Operation		`NUM = 15`
		`X = X + 1`

Input/Output Operation		`input NUM`
		`output "The Number is ", NUM`

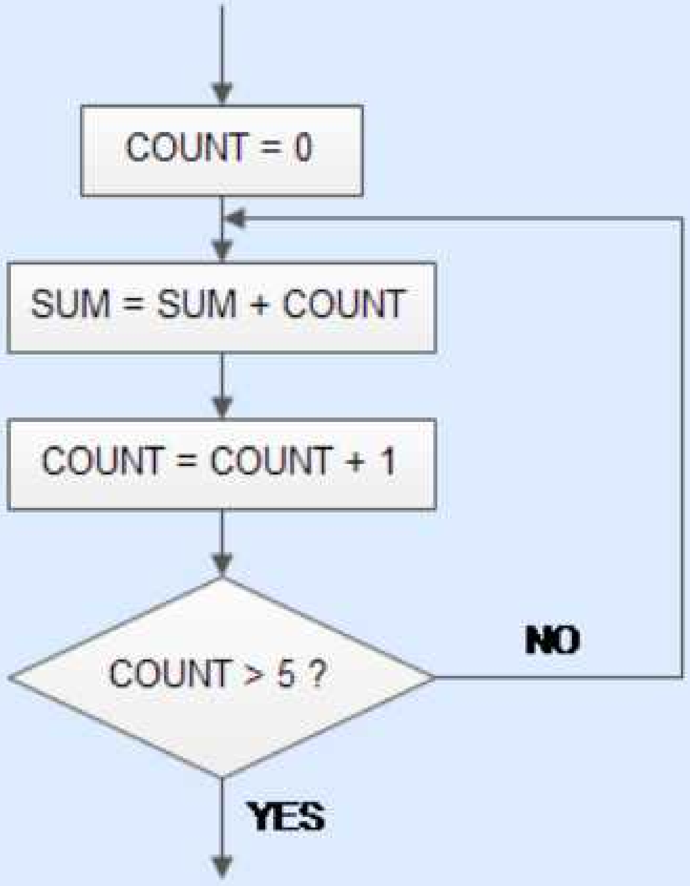
Operation	Flowchart Example	Pseudocode Example

Decision Operation	 MAX > 0?
	NO
	YES
	output "positive"
	output "not positive"
		```
		if MAX > 0 then
		  output "positive"
		else
		  output "not positive"
		end if
		```

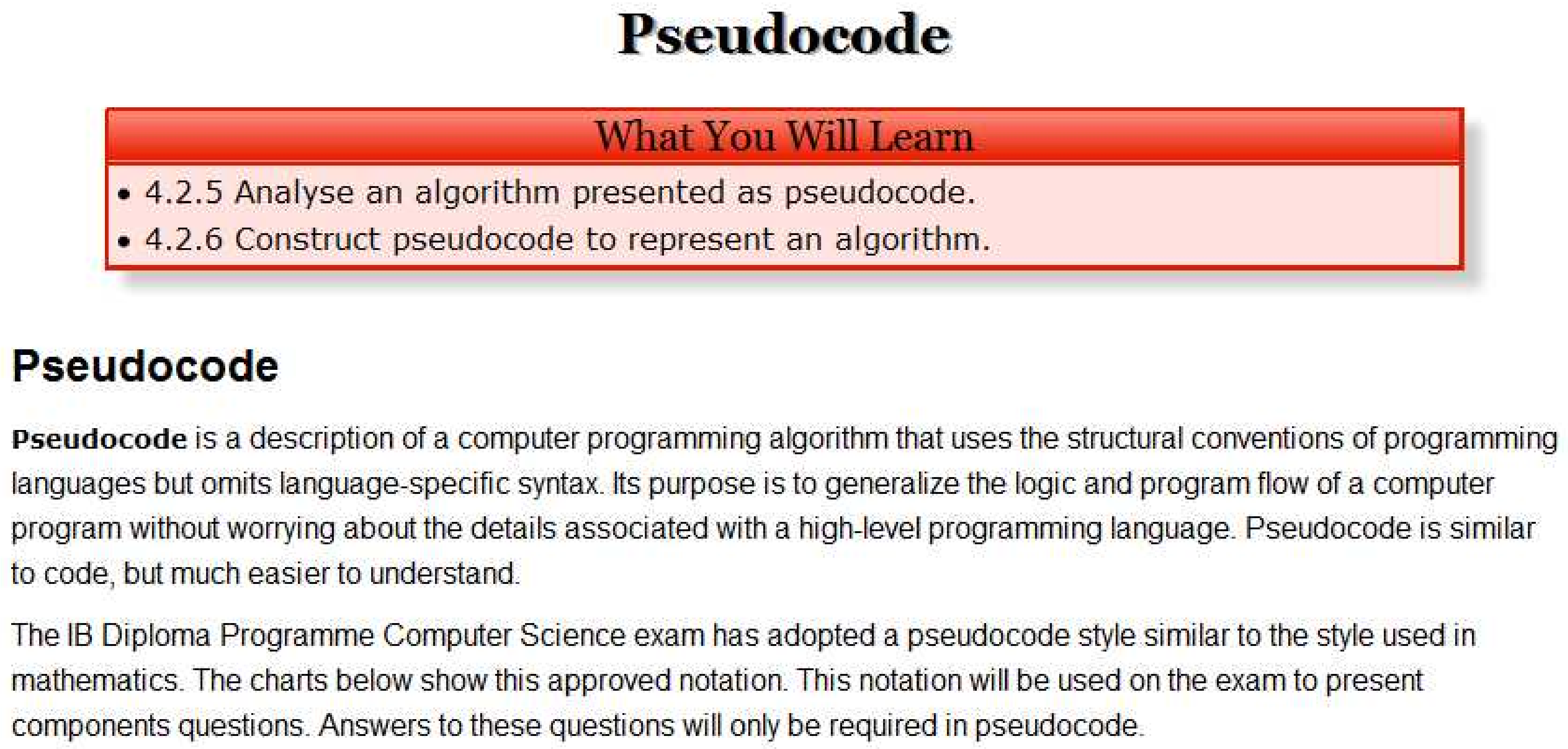
Operation	Flowchart Example	Pseudocode Example

while loop		`loop while COUNT < 15`
		`   COUNT = COUNT + 1`
		`end loop`

Operation	Flowchart Example	Pseudocode Example

for loop	 COUNT = 0
	SUM = SUM + COUNT
	COUNT = COUNT + 1
	COUNT > 5 ?
	NO
	YES
		`COUNT = 0`
		`loop COUNT from 0 to 5`
		`    SUM = SUM + COUNT`
		`end loop`

Part 2: Pseudocode



Pseudocode

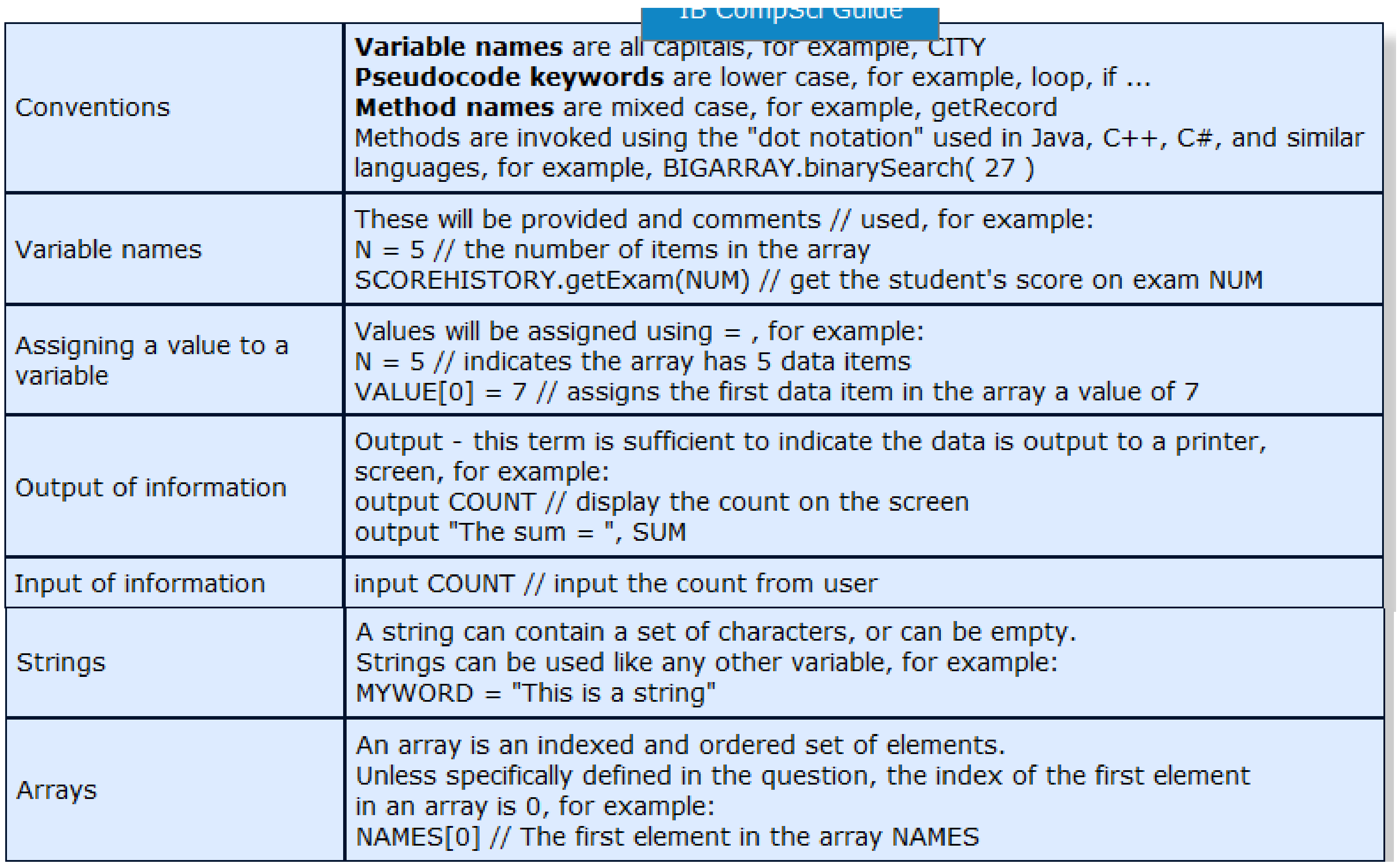
What You Will Learn

- 4.2.5 Analyse an algorithm presented as pseudocode.
- 4.2.6 Construct pseudocode to represent an algorithm.

Pseudocode

Pseudocode is a description of a computer programming algorithm that uses the structural conventions of programming languages but omits language-specific syntax. Its purpose is to generalize the logic and program flow of a computer program without worrying about the details associated with a high-level programming language. Pseudocode is similar to code, but much easier to understand.

The IB Diploma Programme Computer Science exam has adopted a pseudocode style similar to the style used in mathematics. The charts below show this approved notation. This notation will be used on the exam to present components questions. Answers to these questions will only be required in pseudocode.


IB CompSci Guide

Conventions	**Variable names** are all capitals, for example, CITY **Pseudocode keywords** are lower case, for example, loop, if ... **Method names** are mixed case, for example, getRecord Methods are invoked using the "dot notation" used in Java, C++, C#, and similar languages, for example, BIGARRAY.binarySearch(27)

Variable names	These will be provided and comments // used, for example:
	N = 5 // the number of items in the array
	SCOREHISTORY.getExam(NUM) // get the student's score on exam NUM

Assigning a value to a variable	Values will be assigned using = , for example:
	N = 5 // indicates the array has 5 data items
	VALUE[0] = 7 // assigns the first data item in the array a value of 7

Output of information	Output - this term is sufficient to indicate the data is output to a printer, screen, for example: output COUNT // display the count on the screen output "The sum = ", SUM
Input of information	input COUNT // input the count from user
Strings	A string can contain a set of characters, or can be empty. Strings can be used like any other variable, for example: MYWORD = "This is a string"
Arrays	An array is an indexed and ordered set of elements. Unless specifically defined in the question, the index of the first element in an array is 0, for example: NAMES[0] // The first element in the array NAMES

Symbol	Definition	Examples

=	assignment
	is equal to
		X = 4, X = K
		if X = 5 then

>	is greater than	if X > 5 then
>=	is greater than or equal to	if X >= 5 then
<	is less than	if X < 5 then
<=	is less than or equal to	if X <= 5 then
≠	not equal to	if X ≠ 5 then
AND	logical AND	if X = 4 AND y = 5 then
OR	logical OR	if X = 4 OR y = 5 then
NOT	logical NOT	if NOT X = 4 then
mod	modulo or remainder	if X mod 2 = 0 then
div	integer part of quotient integer division	ANS = NUM div 10

if statement	if NUM mod 2 = 0 then
	   output "even number"
	end if

if-else statement	if NOT COUNT = 0 then
	   AVERAGE = TOTAL / COUNT
	   output "Average = " , AVERAGE
	else
	   output "There are no non-zero values"
	end if

cascading if statement	if GR >= 90 then
	   output "A"
	else
	   if GR >= 80 then
	       output "B"
	   end if
	else
	   if GR >= 70 then
	       output "C"
	   end if
	else
	   output "F"
	end if

while loop	`I = 0` `loop while I < 10` `   output I` `   I = I + 1` `end loop`
do-while loop	`I = 0` `loop until I = 10` `   output I` `   I = I + 1` `end loop`
for loop	`loop I from 0 to 10` `   output I` `end loop`