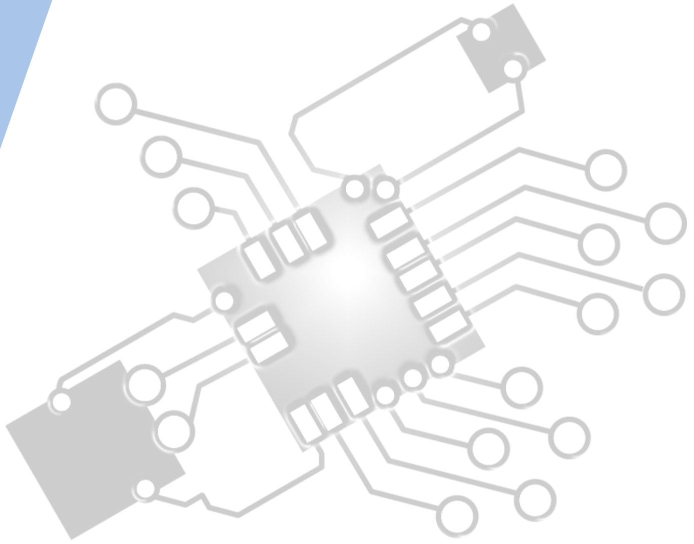
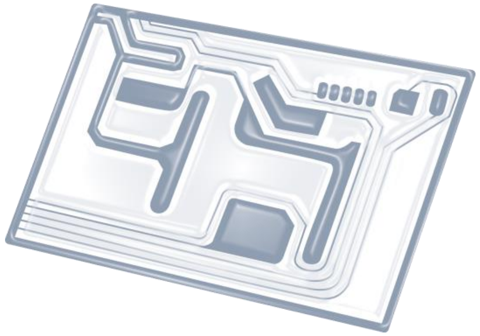
Computational thinking, problem-solving and programming: Introduction to programming

IB Computer Science



Content developed by
Dartford Grammar School
Computer Science Department

HL Topics 1-7, D1-4



1: System design

2: Computer Organisation

3: Networks

4: Computational thinking

5: Abstract data structures

6: Resource management

7: Control

D: OOP

HL & SL 4.3 Overview

Nature of programming languages

4.3.1 State the fundamental operations of a computer

4.3.2 Distinguish between fundamental and compound operations of a computer

4.3.3 Explain the essential features of a computer language

4.3.4 Explain the need for higher level languages

4.3.5 Outline the need for a translation process from a higher level language to machine executable code

Use of programming languages

4.3.6 Define the terms: variable, constant, operator, object

4.3.7 Define the operators =, ., <, <=, >, >=, mod, div

4.3.8 Analyse the use of variables, constants and operators in algorithms

4.3.9 Construct algorithms using loops, branching

4.3.10 Describe the characteristics and applications of a collection

4.3.11 Construct algorithms using the access methods of a collection

4.3.12 Discuss the need for sub-programmes and collections within programmed solutions

4.3.13 Construct algorithms using predefined sub-programmes, one-dimensional arrays and/or collections

1: System design

2: Computer Organisation

3: Networks

4: Computational thinking

5: Abstract data structures

6: Resource management

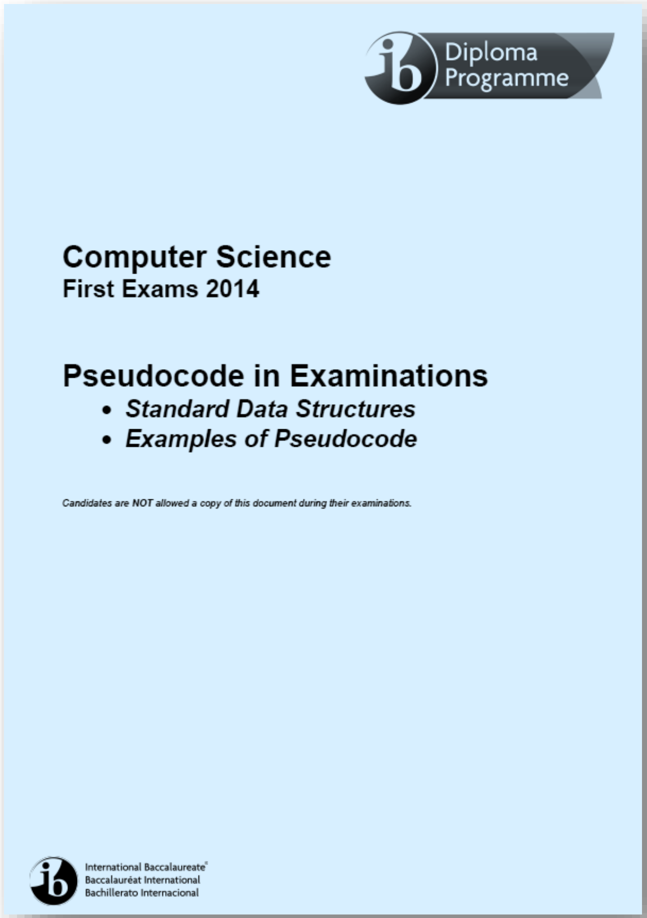
7: Control

D: OOP

Topic 4.3.13

Construct algorithms using predefined **sub-programmes**, **one-dimensional arrays** and/or **collections**

Remember: IB pseudo code



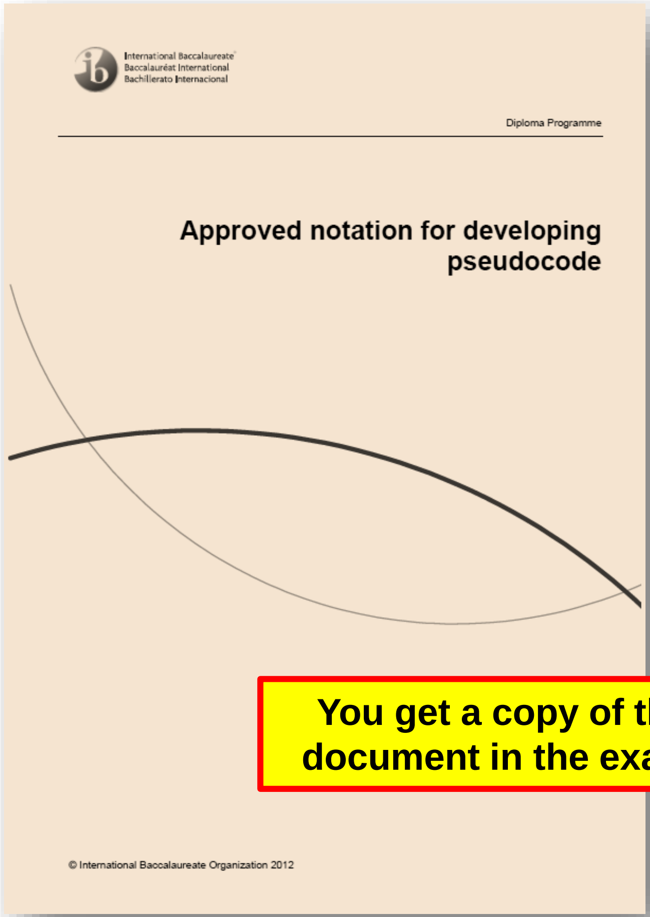
Diploma Programme

Computer Science
First Exams 2014

Pseudocode in Examinations
- *Standard Data Structures*
- *Examples of Pseudocode*

Candidates are NOT allowed a copy of this document during their examinations.

International Baccalaureate
Baccalauréat International
Bachillerato Internacional



International Baccalaureate
Baccalauréat International
Bachillerato Internacional

Diploma Programme

Approved notation for developing pseudocode

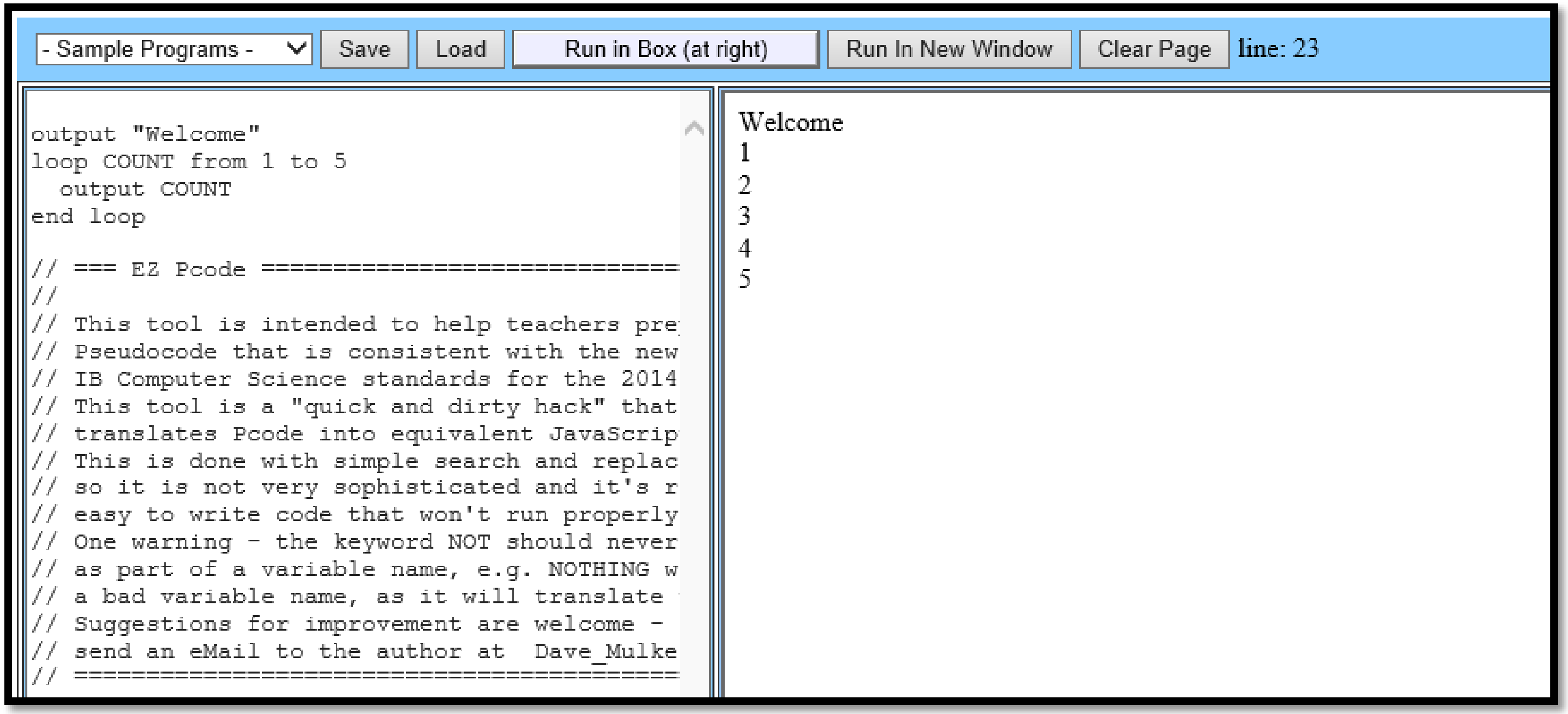
© International Baccalaureate Organization 2012

You get a copy of this document in the exams

Best method: PRACTICE THIS!

Use the *D. Mulkey's* **ONLINE PSEUDO CODE GENERATOR:**

https://dl.dropboxusercontent.com/u/275979/ibcomp/pseduocode/pcode.html



```
output "Welcome"
loop COUNT from 1 to 5
  output COUNT
end loop

// === EZ Pcode =============================
//
// This tool is intended to help teachers pre
// Pseudocode that is consistent with the new
// IB Computer Science standards for the 2014
// This tool is a "quick and dirty hack" that
// translates Pcode into equivalent JavaScrip
// This is done with simple search and replac
// so it is not very sophisticated and it's r
// easy to write code that won't run properly
// One warning - the keyword NOT should never
// as part of a variable name, e.g. NOTHING w
// a bad variable name, as it will translate
// Suggestions for improvement are welcome -
// send an eMail to the author at  Dave_Mulke
// ========================================
```

```
Welcome
1
2
3
4
5
```

Important: only use the methods below

A collection is like a linked-list, but the order of elements is not guaranteed so you can't use `.get(x)` or `.size()` etc.

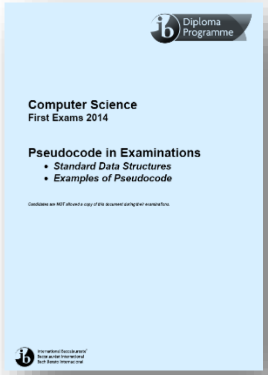
Collection methods in **Pseudocode** are:

• `.addItem( new data item )`	
• `.resetNext( )`	start at beginning of list
• `.hasNext( )`	checks whether there are still more items in the list
• `.getNext( )`	retrieve the next item in the list
• `.isEmpty( )`	check whether the list is empty

AVERAGING AN ARRAY

The array `STOCK` contains a list of 1000 whole numbers (integers). The following pseudocode presents an algorithm that will count how many of these numbers are non-zero, adds up all those numbers and then prints the average of all the non-zero numbers (divides by `COUNT` rather than dividing by 1000).

```
COUNT = 0
TOTAL = 0

loop N from 0 to 999
  if STOCK[N] > 0 then
    COUNT = COUNT + 1
    TOTAL = TOTAL + STOCK[N]
  end if
end loop

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


Diploma Programme

Computer Science
First Exams 2014

Pseudocode in Examinations
- *Standard Data Structures*
- *Examples of Pseudocode*

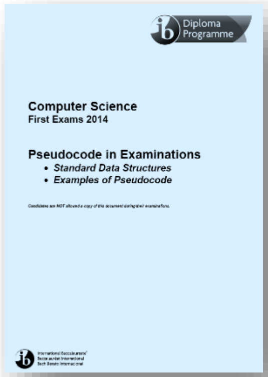
Candidates are NOT allowed a copy of this document during their examinations.

International Baccalaureate
Baccalauréat International
Bachillerato Internacional

COPYING FROM A COLLECTION INTO AN ARRAY

The following pseudocode presents an algorithm that reads all the names from a collection, `NAMES`, and copies them into an array, `LIST`, but eliminates any duplicates. That means each name is checked against the names that are already in the array. The collection and the array are passed as parameters to the method.

```
COUNT = 0   // number of names currently in LIST

loop while NAMES.hasNext()

  DATA = NAMES.getNext()

  FOUND = false
  loop POS from 0 to COUNT-1
    if DATA = LIST[POS] then
      FOUND = true
    end if
  end loop

  if FOUND = false then
    LIST[COUNT] = DATA
    COUNT = COUNT + 1
  end if
end loop
```



Diploma Programme

Computer Science
First Exams 2014

Pseudocode in Examinations
- *Standard Data Structures*
- *Examples of Pseudocode*

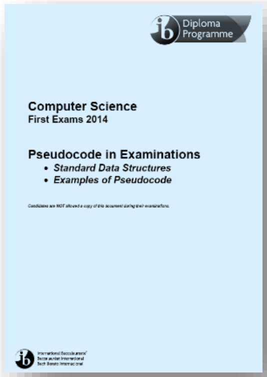
Candidates are NOT allowed a copy of this document during their examinations.

International Baccalaureate
Baccalauréat International
Bachillerato Internacional

FACTORS

The following pseudocode presents an algorithm that will print all the factors of an integer. It prints two factors at a time, stopping at the square root. It also counts and displays the total number of factors.

```
// recall that
//   30 div 7 = 4
//   30 mod 7 = 2

NUM = 140  // code will print all factors of this number
F = 1
FACTORS = 0

loop until F*F > NUM  //code will loop until F*F is greater than NUM

  if NUM mod F = 0 then

    D = NUM div F
    output NUM , " = " , F , "*" , D

    if F = 1 then
      FACTORS = FACTORS + 0
    else if F = D then
      FACTORS = FACTORS + 1
    else
      FACTORS = FACTORS + 2
    end if

  end if

  F = F + 1

end loop

output NUM , " has " , FACTORS , " factors "
```


Diploma Programme

Computer Science
First Exams 2014

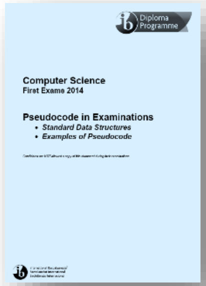
Pseudocode in Examinations
- Standard Data Structures
- Examples of Pseudocode

Content developed by Dartford Grammar School Computer Science Department

COPYING A COLLECTION INTO AN ARRAY IN REVERSE

The following pseudocode presents an algorithm that will read all the names from a collection, `SURVEY`, and then copy these names into an array, `MYARRAY`, in reverse order.

```
// MYSTACK is a stack, initially empty

COUNT = 0 // number of names

loop while SURVEY.hasNext()
  MYSTACK.push( SURVEY.getNext() )
  COUNT = COUNT + 1
end loop

// Fill the array, MYARRAY, with the names in the stack

loop POS from 0 to COUNT-1
  MYARRAY[POS] = MYSTACK.pop()
end loop
```


Diploma Programme

Computer Science
First Exams 2014

Pseudocode in Examinations
- *Standard Data Structures*
- *Examples of Pseudocode*