Pseudo Code

PROGRAMMING ON PAPER FOR IB PAPER 1 EXAMS

2 official pseudo code guides

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.

8 pages
NOT given in exam

Approved notation for developing pseudocode
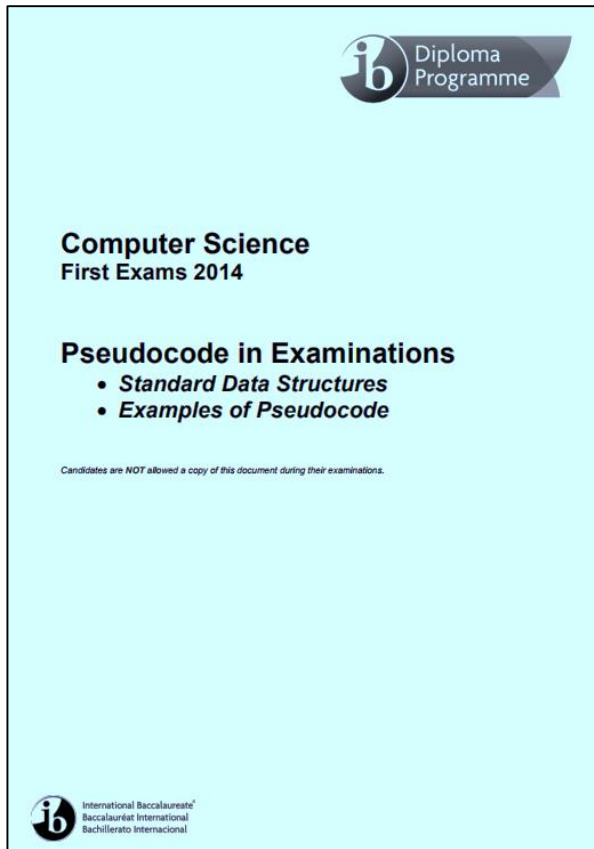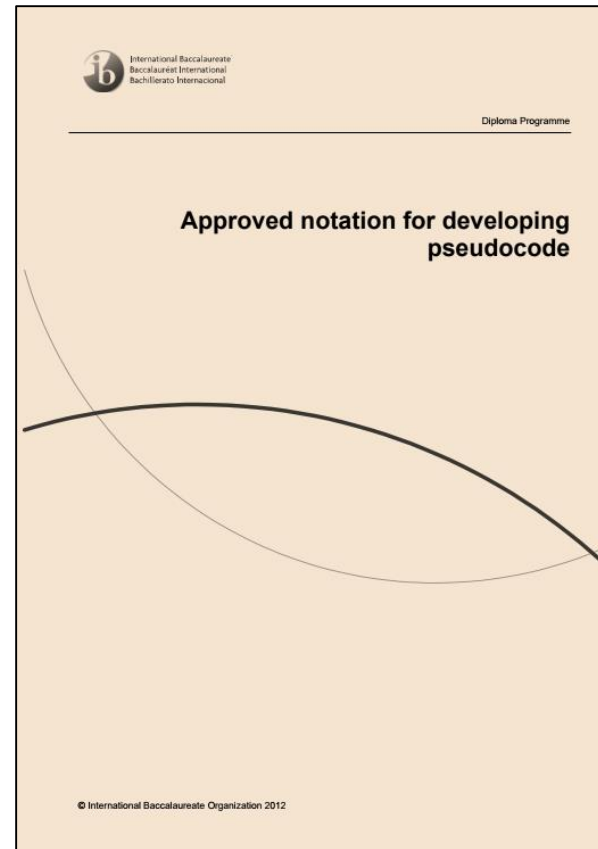
3 pages
Given in exam

Warning!

Pseudo code questions are *never* as overt or obvious as the examples we discuss in these sessions.

These examples are only there to teach you the *skills* you need to answer more complex problems.

Topics 4, 5 and 7 can include pseudo code…

Top pseudo code tips

✓ When possible, start answering a pseudo code question at the top of a page

✓ Write pseudo code in pencil first and then copy into pen

✓ Think about the data types and associated access methods BEFORE writing anything

✓ Think about what control structures (especially loops) are associated with that data type

✓ Be sure to return or output something at the end of the problem – even if they don't ask for it!

Basic structure of ALL pseudo code questions

```
Declarations / Initialisations

Control structures / Calculations

Output / Return
```

You get marks for individual sections, not the final output.

This means you could well get 7/8 even if your final output is not right.

T1: Print out contents of a linear array

You have an array of names called STUDENTS that contains 20 String values.

Write an algorithm that will print out all the student names.

```
1  loop POS from 0 to 19
2       output STUDENTS[POS]
3  end loop
```

T2: Print out contents of a collection

You have a collection of names called SLIST that contains an unknown number of values.

Write an algorithm that will print out all the student names.

```
1 loop while SLIST.hasNext()
2     output SLIST.getNext()
3 end loop
```

T3: Calculate min/max/average of linear array

You have an array of 7 integers called HORSES.

Output the minimum, maximum and average value of the values in the array.

```
1  MAX = HORSES[0]
2  MIN = HORSES[0]
3  TOTAL = 0
4  AVERAGE = 0
5
6  loop NUM from 0 to 6
7      if HORSES[NUM] > MAX then
8          MAX = HORSES[NUM]
9      else if HORSES[NUM] < MIN then
10         MIN = HORSES[NUM]
11     end if
12     TOTAL = TOTAL + HORSES[NUM]
13 end loop
14 AVERAGE = TOTAL / HORSES.length
15
16 OUTPUT MIN
17 OUTPUT MAX
18 OUTPUT AVERAGE
```

T4: Calculate min/max/average of a collection

You have collection of double values called HEIGHTS.

Output the minimum, maximum and average value of the values in the collection.

```
 1 MIN = HEIGHTS.getNext()
 2 MAX = MIN
 3 TOTAL = 0
 4 AVERAGE = 0
 5 COUNT = 0
 6
 7 HEIGHTS.resetNext()
 8 loop while HEIGHTS.hasNext()
 9     TEMP = HEIGHTS.getNext()
10     if TEMP < MIN then
11         MIN = TEMP
12     else if TEMP > MAX then
13         MAX = TEMP
14     end if
15     TOTAL = TOTAL + TEMP
16     COUNT = COUNT + 1
17 end loop
18 AVERAGE = TOTAL / COUNT
19
20 output MIN
21 output MAX
22 output AVERAGE
```

Remember the 5 access methods for collections:
`.getNext()`
`.resetNext()`
`.addItem()`
`.isEmpty()`
`.hasNext()`

T5: Find a value in an array

You have a char array of 26 letters called ALPHA containing all the letters of the English alphabet. There are no duplicate letters.

Return the location (index) of the letter 'C'.

```
1  LOCATION = -1
2
3  loop COUNT from 0 to 25
4      if ALPHA[COUNT] == 'C' then
5          LOCATION = COUNT
6      end if
7  end loop
8
9  return LOCATION
```

Hint 1:
In pseudo code, `==` can be used for int, double, char, Boolean or Strings. There is no `.equals()` method like in Java!

T6: Find a value in a collection

You have a list of students called SLIST.

Search the list for a student called "Sam".

The algorithm should be as efficient as possible.

```
1  FOUND = false
2
3  loop while SLIST.hasNext() and FOUND == false
4      if SLIST.getNext() == "Sam" then
5          FOUND = true
6      end if
7  end loop
8
9  output FOUND
```

Hint 1:
When efficiency is called upon, a flagged loop that can stop early if the value is found is ALWAYS more efficient

Hint 2:
Always output/return a value, even if it is not explicitly asked for