



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

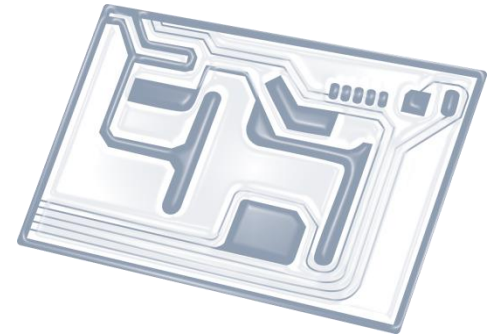
IB Computer Science



*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

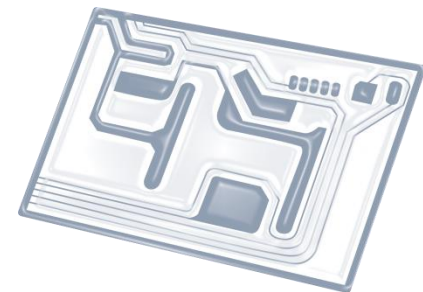
6: Resource management



7: Control

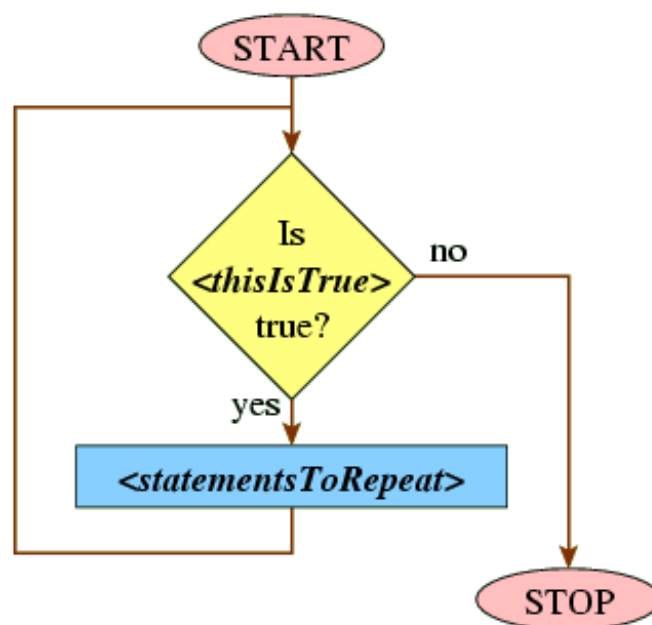
D: OOP





Topic 4.3.9

Construct algorithms using **loops** and **branching**

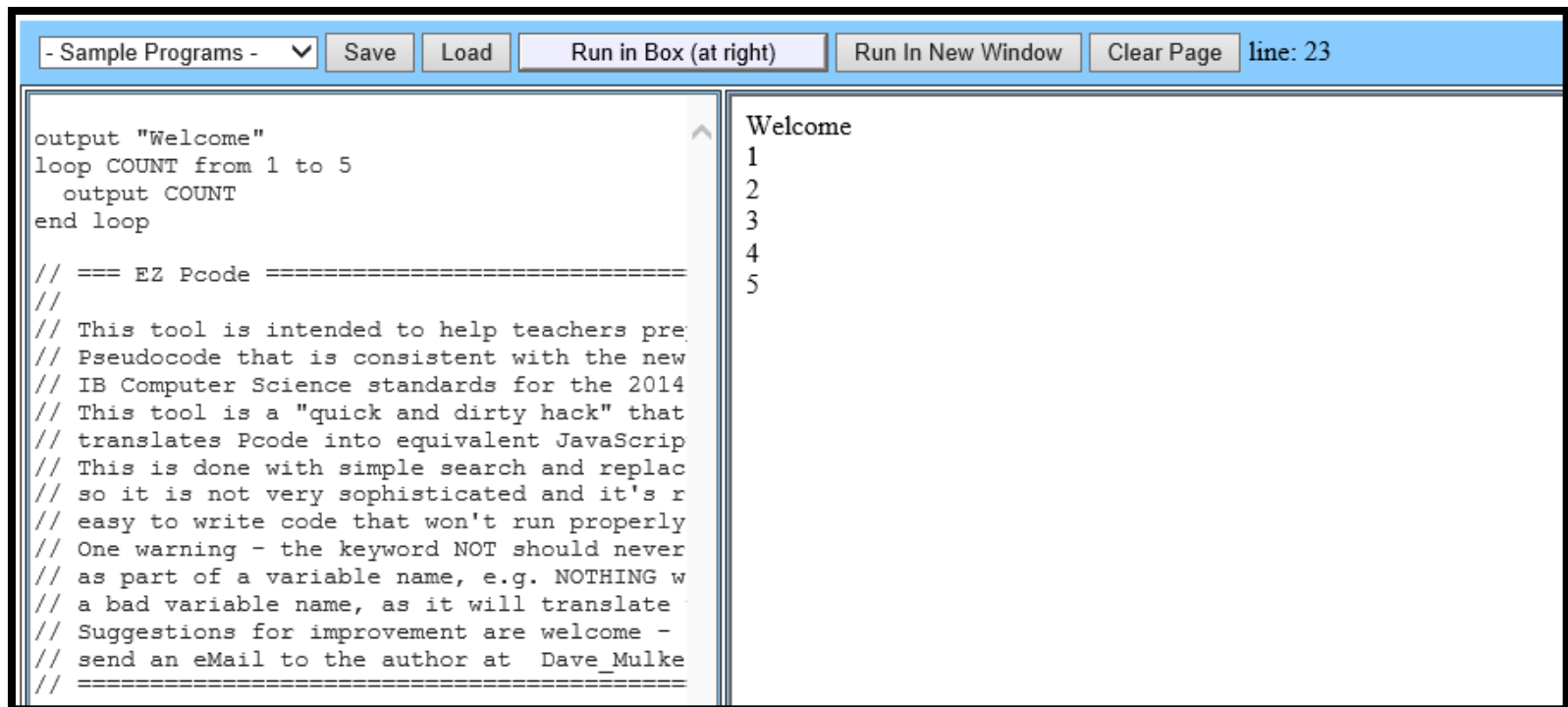


Links with
D.3.6 & 7
(Paper 2)

Best method: **PRACTICE THIS!**

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

<https://dl.dropboxusercontent.com/u/275979/ibcomp/pseudocode/pcode.html>

A screenshot of a web-based pseudo code generator. The interface has a light blue header with a dropdown menu set to '- Sample Programs -', and buttons for 'Save', 'Load', 'Run in Box (at right)', 'Run In New Window', and 'Clear Page'. A 'line: 23' indicator is on the right. The main area is split into two panes. The left pane contains pseudo code: 'output "Welcome"', 'loop COUNT from 1 to 5', ' output COUNT', 'end loop', followed by a large block of comments starting with '// === EZ Pcode ==='. The right pane shows the output: 'Welcome' followed by a vertical list of numbers 1 through 5.

```
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

Practice code that uses:

- IF / ELSE
- Boolean conditions, e.g. **WHILE** `list.hasNext()`

```
int num = 5;

if (num < 3) {
    System.out.println("Under 3");
}
else if (num == 3) {
    System.out.println("Equals 3");
}
else {
    System.out.println("Over 3");
}
```

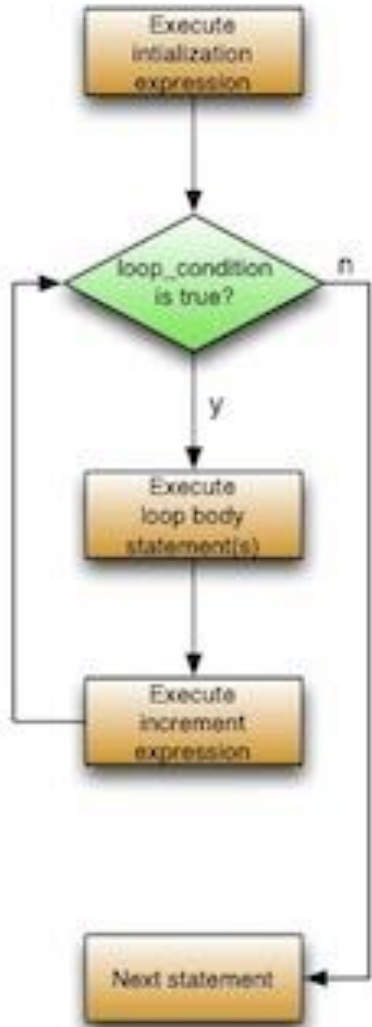
Practice code that uses:

- FOR loops
- WHILE loops

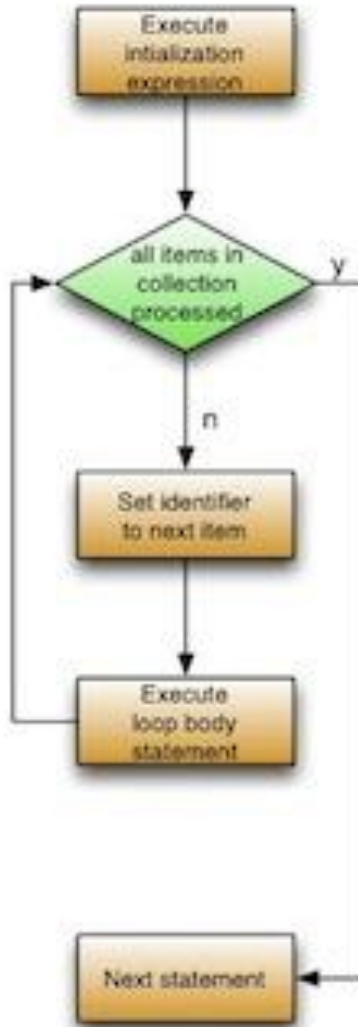
```
public static void main(String[] args) {  
  
    int loopVal;  
    int end_value = 11;  
    int addition = 0;  
  
    for (loopVal = 1; loopVal < end_value; loopVal++) {  
  
        addition = addition + loopVal;  
    }  
  
    System.out.println("Total = " + addition);  
}
```

```
// This part is for Java 2 only  
/* Vector v = new Vector(table.values());  
   Collections.sort(v);  
   Enumeration list = v.elements(); */  
  
// And this is for earlier versions  
Enumeration list = table.elements();  
  
// Common code  
StringBuffer outFileNames =  
    new StringBuffer(inFile.getName());  
outFileNames.setCharAt(0, 'C');  
outFileNames.setCharAt(1, '_');  
  
String outFile = outFileNames.toString();  
try {  
    FileWriter out =  
        new FileWriter(outFile, false);  
  
    while(list.hasMoreElements()) {  
        Word temp = (Word)list.nextElement();  
        out.write(temp.toString() + NLINE);  
    }  
    out.close(); // finished  
} catch(IOException err) {  
    System.err.println("Error in WordCount:count()" +  
        + " outfile:" + NLINE + err.getMessage());  
}
```

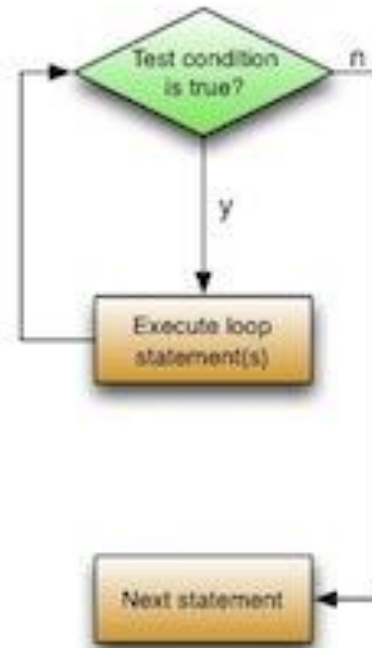
The numerical for loop



The collection-based for loop



The while loop



The do-while loop

