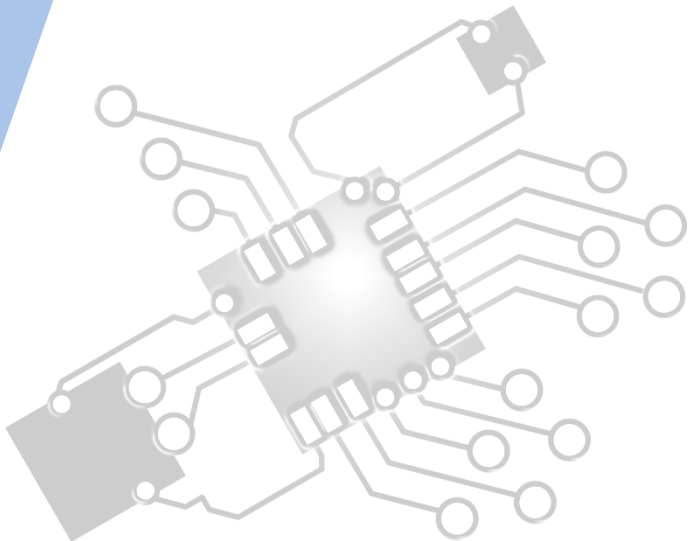




Objects as a programming concept

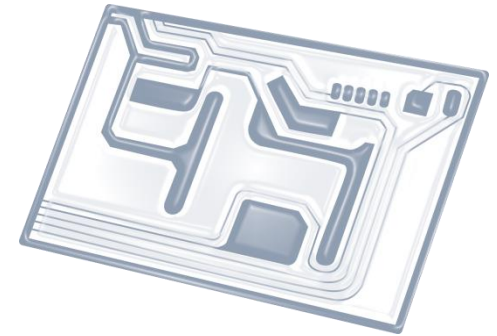
IB Computer Science



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



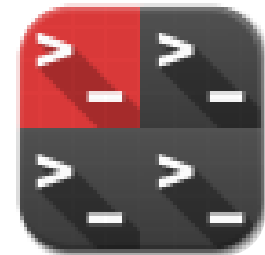
5: Abstract data structures



6: Resource management



7: Control



D: OOP

HL & SL D.3 Overview

D.3 Program development

D.3.1 Define the terms: class, identifier, primitive, instance variable, parameter variable, local variable

D.3.2 Define the terms: method, accessor, mutator, constructor, signature, return value

D.3.3 Define the terms: private, protected, public, extends, static

D.3.4 Describe the uses of the primitive data types and the reference class string

D.3.5 Construct code to implement assessment statements

D.3.6 Construct code examples related to selection statements

D.3.7 Construct code examples related to repetition statements

D.3.8 Construct code examples related to static arrays

D.3.9 Discuss the features of modern programming languages that enable internationalization

D.3.10 Discuss the ethical and moral obligations of programmers



1: System design

2: Computer Organisation



3: Networks

4: Computational thinking



5: Abstract data structures

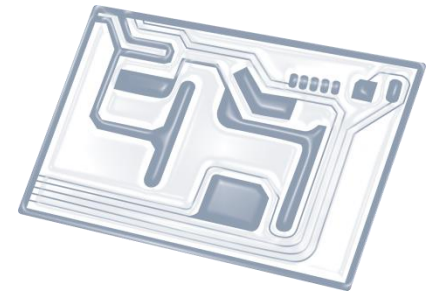
6: Resource management



7: Control

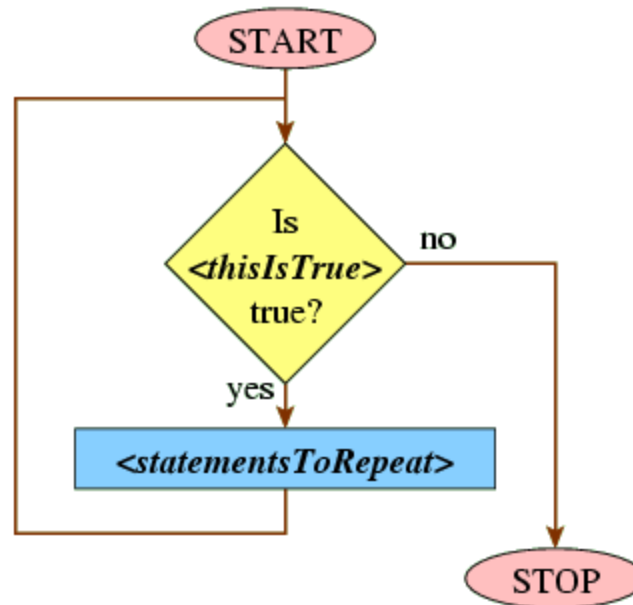
D: OOP





Topic D.3.7

Construct code examples related to **repetition** statements



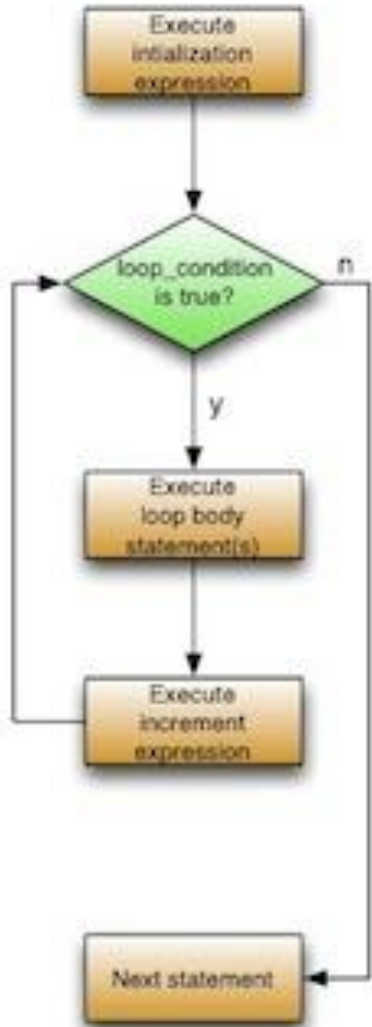
Practice code that uses:

- FOR loops
- WHILE loops

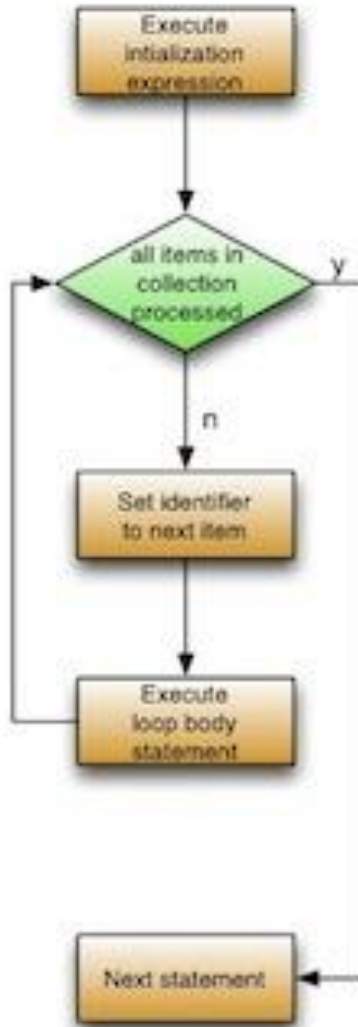
```
public static void main(String[] args) {  
  
    int loopVal;  
    int end_value = 11;  
    int addition = 0;  
  
    for (loopVal = 1; loopVal < end_value; loopVal++) {  
  
        addition = addition + loopVal;  
    }  
  
    System.out.println("Total = " + addition);  
}
```

```
// 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());  
}
```

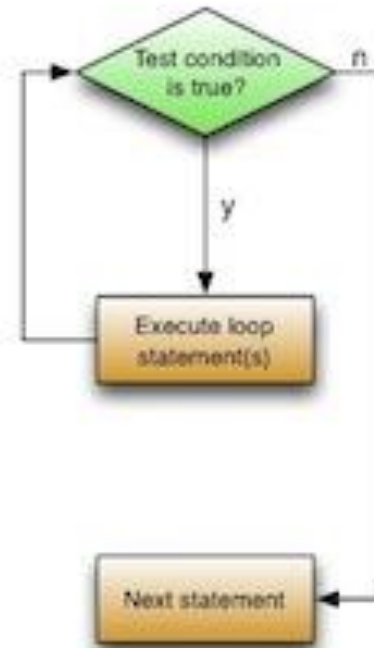
The numerical for loop



The collection-based for loop



The while loop



The do-while loop

