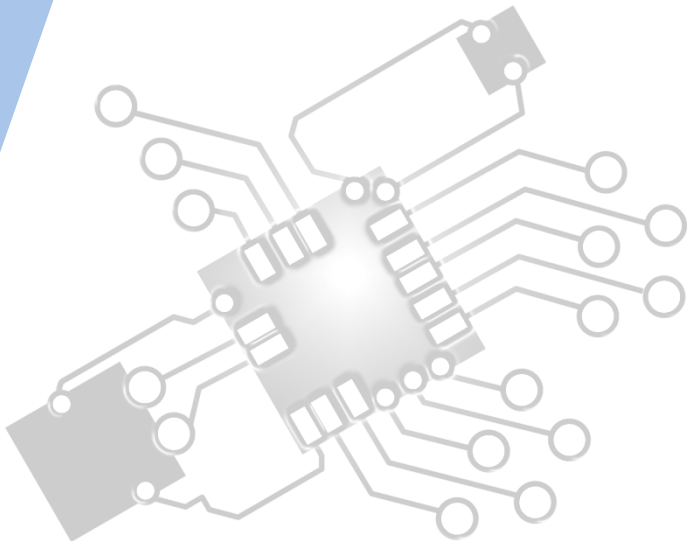




Computational thinking, problem-solving and programming: Introduction to programming

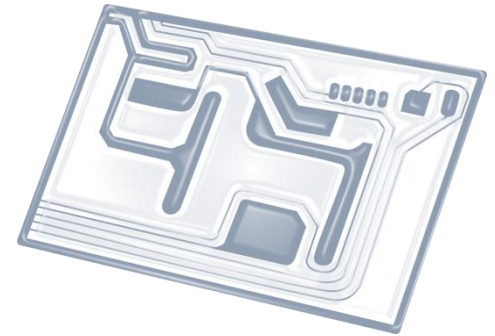
IB Computer Science



*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

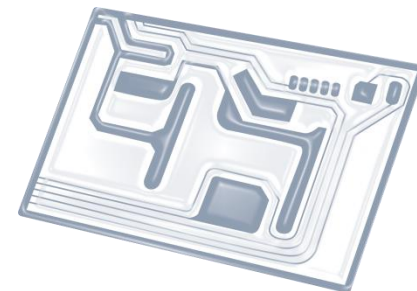
6: Resource management



7: Control

D: OOP





Topic 4.3.4

Explain the **need** for higher level languages

```
#include <stdio.h>
int main(void)
{
    int count;

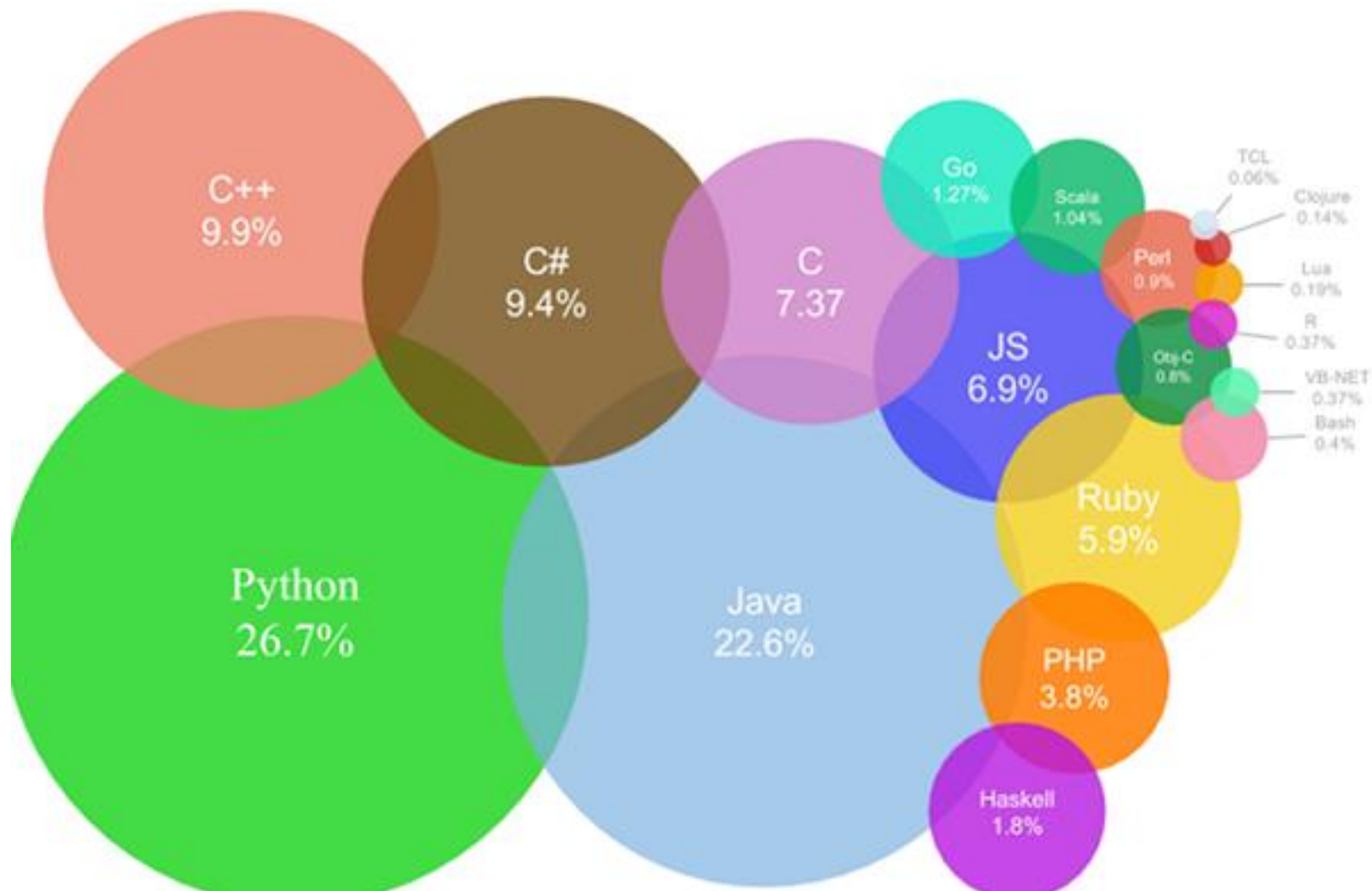
    for (count = 1; count <= 500; count++)
        printf("I will not throw paper airplanes in class.");

    return 0;
}
```

MYEND 10-3



Most popular languages (2016)



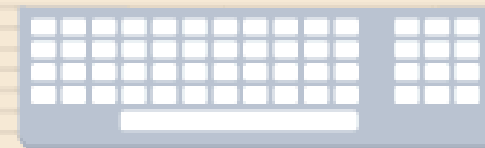
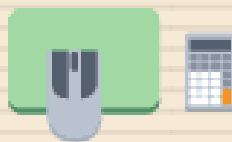
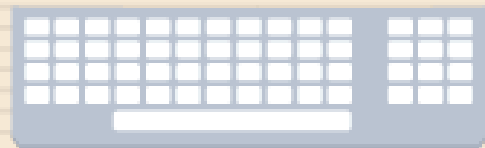
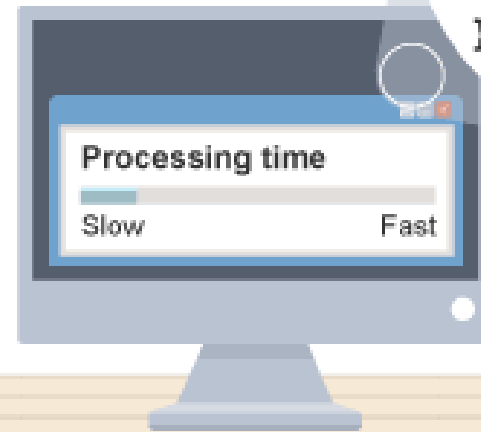
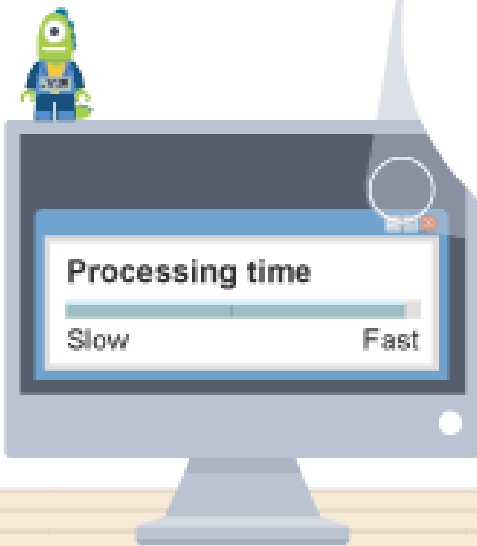
Low level vs High level language

Low level

```
1010010110111010  
1001110110000111  
0001110010110001  
1011010110111010  
0000111001010111  
1001110010011101
```

High level

```
sale_price = 1.66  
if (sale_price > 2) {  
    discount = 0.1  
}  
else {  
    discount = 0.05  
}
```



Why do we **need** high level languages?

- **High level language** = similar to human language (like English)
- **Low level language** = close to the binary code used to actually process the instruction.
- As **human needs for computer systems have expanded**, it is necessary to abstract from the basic operations of a computer.
- It would **take far too long** to write the type of systems needed today in machine code.

Comparison: C vs Java vs Python

- C

```
#include <stdio.h>

int main(int argc, char ** argv)
{
    printf("Hello, World!\n");
}
```

- Java

```
public class Hello
{
    public static void main(String argv[])
    {
        System.out.println("Hello, World!");
    }
}
```

- now in Python

```
print "Hello, World!"
```