

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



2: Computer Organisation





3: Networks

4: Computational thinking





5: Abstract data structures

6: Resource management













Topic 4.3.1

State the **fundamental operations** of a computer





Fundamental operations

All CPUs have sets of instructions, also called the fundamental operations, that enable commands to be processed.

The four most fundamental operations are:

- **√ADD**
- **✓ COMPARE**

✓ **RETRIEVE** (sometimes called **LOAD**)

✓ STORE (sometimes called SAVE)



Fundamental vs Complex

An example of fundamental instructions:

Examples of complex instructions:

LOAD register 34AB39 ADD 29 STORE result COMPARE result to register 4

Find the biggest number in an array

Sort the names alphabetically



Example of Machine Code (using Fundamentals)

| TKP 16514 PUSH_HL 229 | 16544 LD BC NN (169) COLL NN | 1 29 % 205 ⊴0 <i>≤4</i> | CP N | 254 148 |
|--|---------------------------------|----------------------------|------------------|------------|
| (130) PUSH BC 197 | (64) DEC HL | 43 | CALL STR | 48 26 |
| (64) CALL NN 205 187 2 | LDCN | 14 38 | CP N | 203 107 04 |
| LDBH 68 | CALL NN | 205 130 64 | JP NC DIS | 48 19 |
| LDCL 77 | INC HL | 35 | PUSH AF | 245 |
| | 16557 LD A (HL) | 126 | CALL ALIST | 205 141 66 |
| INC D 20 | SUB N | 214 28 | POP AF | 241 |
| JE 2 DID 40 247 FOLL NN - 205 100 7 | | | CP N | 254 0 |
| UD A (HL)126 | LUUN VODA | 14 11 | JP Z DIS | 40 10 |
| POP BC 193 | ADD C | 129 | PUP BU DOD UI | 193 |
| PUSH BC 197 | DJNZ DIS | 16 253 | INAC | 121 |
| CP C 185 | ADD N | 198 97 | CP N | 254 1 |
| JP Z DIS 40 6 | DEC HL | 43 | JP Z DIS | 40 5 |
| | SUB (HL) | 150 | POP AF | 241 |
| DOD DC 192 | MOVE | | JR DIS | 24 218 |
| TO DC 153 TO BIS 24 231 | | 123 | POP BC | 193 |
| POP BC 193 | НШИ Н (НС | 245 | PUP HL | 225 |
| POP HL 225 | PUSH HI | 273 | FUF NF TNC HI | 241 25 |
| LD(HL) A 119 | PUSH BC | 197 | DINZ DIS | 16 211 |
| RET 201 | CP N | 254 63 | RET | 201 |
| KYBD | JP C DIS | 56 30 | 40 MONT | |