# Homework 3 Programming language classification Answers

1. (a) State **two** advantages of writing a program in assembly code over writing a program in machine code. [2]

easier to remember mnemonics than strings of binary digits

less likely to make mistakes in the operand using decimal instead of binary numbers

quicker to write programs

easier to understand, debug and maintain

machine code more suited to programming by physically setting switches to load instructions

(b) State **two** advantages of writing a program in a high-level language over writing a program assembly code. (Do not repeat any answers from part (a)). [2]

High level of abstraction enables programmers to translate algorithms more or less directly into program code

Frees programmers from having to know how the computer executes every instruction

Programs much shorter because each instruction is translated into many machine code instructions

Can chose a language best suited to the application, e.g. imperative or declarative

Many built-in functions and libraries available

(c) State **two** reasons why a programmer might choose to write a program in assembly code rather than in a high-level language. [2]

Very little memory space available, e.g. embedded systems

Program needs to execute as fast as possible, e.g. in real time

Program needs to manipulate individual bits and bytes

2. Some of the assembly language instructions supported by a simple microprocessor are:

|  |  |
| --- | --- |
| **Instruction** | **Meaning** |
| LDA | Load the value stored in memory location specified by the operand into the accumulator |
| STO | Store the value in the accumulator in memory location specified by the operand |
| ADD | Add the value specified in the operand to the value in the accumulator |
| CMP | Compare the contents of the accumulator with the contents of the location specified by the operand |
| BLT | Jump to the address held in the operand if the accumulator held the lesser value in the last comparison |
| BGT | Jump to the address held in the operand if the accumulator held the greater value in the last comparison |
| JMP | Jump to the address held in the operand |
| STOP | Stop |

(a) Write into the table below the opcode and the operand in the following instruction:

STO 8

|  |  |
| --- | --- |
| **Operand** | 8 |
| **Opcode** | STO |

[1]

(b) Write an assembly language program using the instructions given above, equivalent to the high-level language statement below. Halt the program after execution of the statement(s).

Assume that the value 1 is held in memory address 9, variable a in address 10 and variable b in address 11.

Comment each line of code to say what it does.

IF (a >= b) THEN

b = b + 1

ENDIF [5]

1 LDA 1010 ; load a

2 CMP 1011 ; compare with b

3 BLT 0111 ; branch to instruction at 7 if a < b

4 LDA 1001 ; load 1 from address 9

5 ADD 1011 ; add b to 1 in accumulator

6 STO 1011 ; store result back in b at address 11

7 STOP

or,

1 LDA 1011 ; load b

2 CMP 1010 ; compare with a

3 BGT 0111 ; branch to instruction at 7 if b > a

5 ADD 1001 ; add 1 held in address 9 to b in accumulator

6 STO 1011 ; store result back in b at address 11

7 STOP

[Total 12 marks]