1.2.4 High vs. Low Level Languages

1. A digital coffee making machine uses an operating system written in assembly language.

a) Discuss the differences between assembly code and high-level languages. You should refer to:

- the advantages and disadvantages of writing programs in assembly code rather than a high-level language
- when each approach might be used
- why the coffee machine was programmed in assembly code.

[9 marks]

Knowledge

- **Stores the value 10 Assembly code uses mnemonics to represent machine code instructions/opcodes.**
- **High level languages use more natural/mathematical notation.**
- **Assembly code consists of simple instructions**
- **As such many more lines of assembly code are required to perform the same task as a few lines of a high level language.**
- **Assembly code is specific to the instruction set of a given processor. High Level languages are not architecture specific.**

Application

- **Assembly code allows the programmer to choose the exact instructions so they can write code that is highly efficient.**
- **It also allows them to have direct control of how memory is used via addressing modes.**
- **Direct control of hardware.**
- **High level language compilers have optimisers that can also try and do this (and in some cases may outperform a human writing in assembly code).**
- **As high level code is more intuitive and easier to read it is easier to follow, debug and build as part of a team. It can also be written in a much shorter time frame.**
- **The high level code can be recompiled for different architectures.**
- **High level languages come in a variety of paradigms so programmers can choose according to the problem/their preference.**

Evaluation

- **Assembly language is best suited to situations such as: compilers or interpreters don't exist for the target CPU i.e. embedded systems**
- **highest possible performance is critical**
- **memory is very limited.**
- **For larger projects which don't fall under the constraints above high level languages are likely to be preferable.**