Languages

Q1. Describe **two** ways that a high-level language differs from a low-level language.

(4)

1 ....

....

....

....

2 ....

....

....

....

Q2. A team of programmers is creating the code for an alarm system. The system uses a high-level programming language for the touchscreen graphical user interface and a low-level language for the control unit that monitors the sensors and triggers the alarm.

Discuss the characteristics of high-level languages and low-level languages that make them appropriate for the team of programmers to code these uses.

Your answer should consider:

- the purpose of the system
- the advantages of high-level languages
- the advantages of low-level languages.

(6)

....

....

....

....

....

....

....

....

....

....

....

....

ANSWERS

1. Any of the differences in the table…….

Aspect	High-Level Language	Low-Level Language
Ease of Understanding	Easy for programmers to understand. Easier to learn; easier to read.	Tough for humans to comprehend so harder to learn.
Level of abstraction	High level of abstraction which contributes to ease of use and understanding	Low level of abstraction which makes it easier to work directly with the hardware
Memory Efficiency	Less memory efficient.	High memory efficiency.
Debugging	Easier debugging process.	More complex debugging.
Maintenance	Simpler to maintain.	Complex maintenance comparatively.
Portability	Portable across platforms.	Non-portable; machine-dependent.
Translation Method	Requires a compiler or interpreter.	Requires an assembler for translation.
Common Usage	Widely used for programming.	Less common. Still used for programming of embedded systems and device drivers

2. Use the table and apply them to the context of the question.

Eg. The alarm system has two distinct components:

Graphical User Interface (GUI): This part deals with the touchscreen interface that users interact with. It requires ease of development, user-friendliness, and abstraction.

Control Unit for Sensors and Alarms: This component handles real-time monitoring, processing data from sensors, and alarm triggering. It demands efficiency, direct hardware control, and performance.

For the GUI a high-level language is appropriate for the team to use because it will be faster to code and debug. High-level languages are independent of specific hardware. Theycan run on various computers without modification. For the GUI, this flexibility is essential.

High-level languages abstract low-level details (like memory management) and emphasize problem-solving logic. This abstraction simplifies GUI development.

High-level languages highlight syntax errors during runtime, aiding programmers in identifying and fixing issues efficiently.

The control unit is an embedded system. A low level language is more appropriate because it will take up less memory. It will run faster which in a real-time system is essential.