2. Language translators

5.2 Language Translators	
Candidates should be able to:	Notes and guidance
Show understanding of the need for: • assembler software for the translation of an assembly language program • a compiler for the translation of a high-level language program • an interpreter for translation and execution of a high-level language program	
Explain the benefits and drawbacks of using either a compiler or interpreter and justify the use of each	
Show awareness that high-level language programs may be partially compiled and partially interpreted, such as Java	
Describe features found in a typical Integrated Development Environment (IDE)	Including: • for coding, including context-sensitive prompts • for initial error detection, including dynamic syntax checks • for presentation, including prettyprint, expand and collapse code blocks • for debugging, including single stepping, breakpoints, i.e. variables, expressions, report window

- Assembler
  - Translates assembly program into machine code
  - Either store the translated program directly in main memory for execution
  - Or store the translated program on a storage medium to be used later
  - Every different type of computer/chip has its own machine code and assembly language
- Compiler
  - Used when development complete (ready for distribution)
  - Reads the source code and reports all errors
  - Run/test the program multiple times without recompilation
  - The compiled file executed faster
  - Produce an executable file (which is no longer compiler dependent)

    - Cross-compilation, the program can be compiled to run on different platforms
    - Executable file runs faster than interpreter
- Interpreter
    - Used during development
    - Test/run incomplete program
    - Debugging is easier
    - Because errors are reported and can be corrected as they are found
    - Can run partially complete program
    - Change the program and see the effect in real-time
    - Translates an statement and executes it immediately
    - Parts of the program is tested, without all the program code available

High level language may be partially compiled and partially interpreted, such as java

- IDE
    - Coding
        - Context-sensitive prompts
    - Initial error detection
        - Dynamic syntax checks
    - Presentation
        - Pretty-print
        - Expand and collapse code blocks
    - Debugging
        - Single stepping
        - Breakpoints
