

Past-paper walk-through

Types of programming language, translators and integrated development environments ■ 4.2

eXercise

Questions in this set

- 0478/12 N25 Q3 ■ 4 marks
- 0478/12 M25 Q7 ■ 9 marks
- 0478/11 N24 Q6 ■ 10 marks
- 0478/13 N24 Q7 ■ 4 marks
- 0478/11 J23 Q7 ■ 4 marks
- 0478/12 J23 Q5 ■ 9 marks

Reading the mark scheme

- **B1** a mark that stands on its own – the point is either made or not.
- **M1** a *method* mark: the right approach, even if the number is wrong.
- **A1** an *answer* mark, and it usually needs its M mark first.
- **C1** a working mark on the way to the answer.
- **//** separates answers the examiner accepts equally.
- **ecf** = error carried forward: a later part is marked on your own earlier answer.
- **owtte** = “or words to that effect” – the wording need not match.

Question 3

0478/12 N25 ■ Q3 ■ 4 marks

A programmer uses assembly language to create a program for a coffee machine.

(a) Describe what is meant by assembly language. [2]

(b) Identify the translator needed for assembly language. [1]

(c) Tick (✓) **one** box to show which statement is an advantage of using assembly language. [1]

A It is close to human language.

B It is memory efficient.

C It is easy to debug.

D It is machine independent.

Solution 3

(a)

Mark scheme

- Any two from:
 - ▪ Low-level language
 - ▪ Uses mnemonics
 - ▪ Used to communicate directly with the computer hardware

Solution 3

(b)

Mark scheme

- Assembler

Solution 3

(c)

Mark scheme

■ B

Question 7

0478/12 M25 ■ Q7 ■ 9 marks

Programs can be written in a high-level language or a low-level language.

(a) Machine code is one type of low-level language.

Identify **one** other type of low-level language.

Identify the translator that is needed for the type of low-level language you have identified.

Type of low-level language

[2]

(b) A high-level language can be translated into a low-level language, using a compiler or an interpreter.

Describe the advantages of using an interpreter instead of a compiler during software development.

[4]

(c) Integrated development environments (IDEs) provide translators.

Identify **three** other common functions of an IDE.

Question 7

1. 2. 3.

[3]

Solution 7

(a)

Mark scheme

- Language: assembly
- Translator: assembler

Solution 7

(b) Mark scheme

- Any four from
 - Each statement is translated and executed before the next
 - Translator stops when an error is found
 - ... can correct and continue from where error found // real time debugging when/where error found
 - ... does not need to retranslate all the code again
 - Can test sections of the code
 - Can test without all the program code being functional/accurate/complete

Solution 7

(c) Mark scheme

- Any three from:
- Example
 - ■ Code editor
 - ■ Run-time environment
 - ■ Error diagnostics // by example
 - ■ Auto-complete
 - ■ Auto-correct
 - ■ Prettyprint // syntax highlighting

Solution 7

- ■ Collapse/expand block
- ■ Auto-documentation

Question 6

0478/11 N24 ■ Q6 ■ 10 marks

A computer programmer uses assembly language to create a computer program for an embedded system in a washing machine.

(a) State what is meant by an embedded system. [1]

(b) Give the name of the translator that will be used for the program. [1]

(c) The washing machine needs to display error codes on a small screen if there is a problem with the washing machine.

The error codes are stored as binary. The binary numbers are too long to be displayed on the washing machine.

State how the error codes could be reduced in length to be displayed on the screen. [1]

(d) Give **one** benefit to the programmer of using assembly language to write the program. [1]

Question 6

(e) The programmer uses an integrated development environment (IDE) to write the program.

A built-in translator to convert the program into machine code is one way that the IDE helps the computer programmer.

Describe **three** other ways the IDE can help the programmer.

[6]

Solution 6

(a)

Mark scheme

- Any one from:
 - ■ Computer system that is designed to a perform dedicated/single function
 - ■ Computer system that contains a microprocessor (and)/dedicated hardware
 - ■ Computer system that is built into a larger system

Solution 6

(b)

Mark scheme

- Assembler

Solution 6

(c)

Mark scheme

- Any one from:
 - ■ Convert to hexadecimal
 - ■ Convert to denary
 - ■ A character set can be used

Solution 6

(d)

Mark scheme

- Any one from:
 - ■ More control over manipulating the hardware
 - ■ Faster execution for testing than a high-level language
 - ■ They can use machine specific instructions

Solution 6

(e) Worked steps

Six marks are available but **at most four for merely stating features** — so every feature must be paired with **what it does for the programmer**, and the marks come in pairs.

Features that help while **writing**:

- **Code editor** — lets the user enter and amend the program's code.
- **Auto-complete / context-sensitive prompts** — suggests the rest of a keyword or the parameters a routine expects.
- **Auto-correction and dynamic syntax checking** — flags a mistyped keyword as the line is typed rather than at compile time.

Solution 6

- **Prettyprint / auto-indentation and syntax highlighting** — colours keywords and indents blocks so the structure is visible.

Features that help while **running and testing**:

- **Run-time environment** — runs the program from inside the IDE without leaving it.
- **Translator** — an interpreter for line-by-line checking, or a compiler for the finished program.
- **Error diagnostics / report window** — reports the type of error and the line it is on.
- **Breakpoints and single stepping** — stop execution at a chosen line, or run one line at a time, watching the variables change.

Solution 6

Write each as **feature** — **what it does**. A bare list of eight names caps at four marks; four properly explained pairs reach six.

The pairing to notice: the writing features stop errors being made, and the running features find the ones that were. [Mark scheme](#)

- Six from (MAX 4 for stating features):
- Code editor ...
- ... that allows the user to enter and amend code in their program
- Run-time environment
- ... that allows a program to be run and see the outputs of their program
- Error-diagnostic

Solution 6

- ... to show the programmer where there are errors in the program
- Auto-completion
- ... to give the user options/suggestions of key commands to select
- Auto-correction
- ... to correct a command that has a minor misspelling
- Prettyprint
- ... changes the colour of key commands so they are easy to identify

Question 7

0478/13 N24 ■ Q7 ■ 4 marks

A computer programmer uses an integrated development environment (IDE) when creating a computer program.

Explain the purpose of the IDE.

[4]

Solution 7

Mark scheme

Any four from:

- It is a piece of software
- ... used to write/develop/edit code
- ... used test/debug the code
- ... with features such as auto-completion // any suitable example
- To translate the code to low level language/machine code

Question 7

0478/11 J23 ■ Q7 ■ 4 marks

- 7 A programmer uses a low-level language to write a computer program for a vending machine.
- (a) Describe what is meant by a low-level language.

Question 7

- (b) Give **two** reasons why the programmer would choose to write the computer program in a low-level language instead of a high-level language.

Question 7

[2]

Solution 7

(a) Mark scheme

- Any two from:
 - ■
 - Close to the language processed by computers
 - ■
 - May use mnemonics
 - ■
- An example is assembly language/machine code
- 2

Solution 7

(b) Mark scheme

- Any two from:
 - ■
 - Can directly manipulate the hardware
 - ■
 - No requirement for the program to be portable
 - ■
 - Program will be more memory efficient
 - ■

Solution 7

- No requirement for a compiler/interpreter
- ■
- Quicker to execute
- ■
- Can use specialised hardware
- 2

Question 5

0478/12 J23 ■ Q5 ■ 9 marks

5 A programmer writes a computer program using a high-level language.

(a) Tick (✓) **one** box to show which statement is correct about writing computer programs in a high-level language.

A Mnemonics are used to create instructions.

☐

B The computer program is harder to debug than a low-level language program.

☐

C The computer program is machine independent.

☐

D The hardware of the computer can be directly manipulated.

☐

[1]

Question 5

- (b) The programmer uses a compiler to translate the computer program.
 - (i) Describe how the compiler translates the computer program.

Question 5

- (ii) Describe how the compiler reports errors.

Question 5

- (c) The programmer uses an integrated development environment (IDE) to create the computer program.

One function of the IDE is that it has the built-in compiler.

Give **three** other common functions of an IDE.

Question 5

[3]

Solution 5

(a)

Mark scheme

■ ■

■ C

■ 1

Solution 5

(b)(i) Mark scheme

- Any three from:
 - ■
 - It translates the (high-level language) to low-level language/object code/machine code
 - ■
 - It translates all the code before it is executed
 - ■
 - It creates an executable file

Solution 5

■ 3

Solution 5

(b)(ii) Mark scheme

- Any two from:
 - ■
 - It creates an error report after trying to compile
 - ■
 - ... displaying all errors in the code
 - ■
 - ... that require correction before execution can take place
- 2

Solution 5

- 0478/12
- May/June 2023

Solution 5

(c) Mark scheme

- Any three from:
- e.g.
- ■
- Code editors
- ■
- Run-time environment
- ■
- Built-in interpreter

Solution 5

- ■
- Error diagnostics
- ■
- Auto-completion
- ■
- Auto-correction
- ■
- Prettyprint
- 3