

End of topic quiz - Topic 2.5 Programming Languages and Integrated Development Environment

1. Name **one** example of a high-level programming language. [1 mark]

2.

a. What is machine code? [1 mark]

b. Name one type of translator that is provided by an IDE. [1 mark]

c. Name two advantages for writing computer programs using a high-level language. [2 marks]

3.

a. Some languages are translated using a compiler. What is another suitable translator? [1 mark]

b. What are two differences between the two types of translator? [2 marks]

4. How would a programmer use a run-time environment within an IDE? [1 mark]

5.

a. What are three useful features of an IDE? [3 marks]

b. For one of the features you have given in part (a), why is this feature helpful? [1 mark]

6.

a. Computer programs can be written in high-level languages. What are two differences between high-level languages and low level language? [2 marks]

7.

a. How does code completion help programmers? [1 mark]

b. How do debugging tools help programmers? [1 mark]

c. Provide a name and purpose of one other feature from an IDE. [1 mark]

8.

a. Which piece of code relates to which type of programming language below? Circle the correct code [2 marks]

- i. High-level language – Code 1 / Code 2
- ii. Machine code – Code 1 / Code 2

Code 1	Code 2
score = score + 1	0101 0011 1010 0001 0000 0001 0011 0011 1010

b. An interpreter would translate the code between which two types of language? [1 mark]

9. A program is needed to design 3D models that can be used by pupils in schools. The programmer can use either low level language or a high-level language to implement this program. The programmer also needs to choose a type of translator to use. Give the consequences for each choice and advise the programmer on what choices should be made. [5 marks]

10. Victoria creates a program using an Integrated Development Environment (IDE).

Describe two tools or facilities that an IDE commonly provides [4 marks].

Answers

1. Name **one** example of a high-level programming language.

Any suitable language, for example:

- C/C++/C#/Objective C
- Small BASIC/Visual BASIC
- Python
- Java
- Javascript
- Pascal/Delphi
- Any other suitable answer

2.

a. What is 'machine code'?

Binary representation of instructions. In a format that the CPU can process. Made up of opcode and operand.

b. Name one type of translator that is provided by an IDE.

Compiler. Interpreter.

c. Name two advantages for writing computer programs using a high-level language.

Closer to natural English ...

- ... so easier to write
- ... so easier to maintain code

Fewer lines of code ...

- ... so quicker to write
- ... so easier to read

Not machine independent ...

- ... so can run the code on different platforms

3.

a. Some languages are translated using a compiler. What is another suitable translator?

Interpreter

b. What are **two** differences between the two types of translator?

Compiler translates code in one batch
... interpreter translates one line at a time.
Compiler provides a list of all syntax errors
... interpreter stops at the first error.
Compiler translates all code before executing
... interpreter executes each line as soon as it is translated.
Compiler produces a machine code file
... interpreter does not.

Compiled code can be executed again without being re-translated
... interpreted code needs to be translated each time it is run.
Compiled machine code keeps the source code secret
... interpreted programs mean that the source code is visible.

4. How would a programmer use a runtime-environment within an IDE?

This lets the programmer run code from within the IDE. Allows for testing of the code during production.

5.

a. What are **three** useful features of an IDE?

- Built in translator
- Syntax highlighting/colour coding
- Code completion
- Automatic indentation
- Automatic documentation
- Error checking
- Online help
- Debugging tools
- Run-time environment

b. For one of the features you have given in part (a), why is this feature helpful.

e.g.

Translator

Can run the program from the editor

...so can test the program straight away

Syntax highlighting

Colour coding for functions/variables/etc.

...so can read the program/identify errors more easily

Code completion

Will suggest variable/function names while typing

...easier to remember/quicker to type/fewer keying errors

Error checking

Will identify and report back on syntax errors

...making it easier to identify and correct them

Debugging tools

Can step through a program line by line

...making it easier to identify and correct logic errors

Online help

Will provide help on using the IDE, possibly including programming syntax

...making it easier to solve problems when stuck

Run-time environment

Features all the requirements for the program to run

...making it quicker to test the program without installing extra libraries

6.

a. Computer programs can be written in high-level languages. What are two differences between high-level languages and low level language? [2 marks]

HLL is closer to English/Machine code is written in binary

One line of HLL code can map to several lines of machine code

HLL is designed to be read/written by humans, machine code is designed to be read/written by computer systems

HLL can be machine independent/Machine code is machine specific (will only run on that type of system).

7.

a. How does code completion help programmers?

Will suggest variable/function names while typing
...easier to remember/quicker to type/fewer keying errors.

b. How do debugging tools help programmers?

Can step through a program line by line
...making it easier to identify and correct logic errors.

c. Provide a name and purpose of one other feature from an IDE

e.g.

Translator

Can run the program from the editor
...so can test the program straight away

Syntax highlighting

Colour coding for functions/variables/etc.
...so can read the program/identify errors more easily

Error checking

Will identify and report back on syntax errors
...making it easier to identify and correct them

Online help

Will provide help on using the IDE, possibly including programming syntax
...making it easier to solve problems when stuck

Run-time environment

Features all the requirements for the program to run
...making it quicker to test the program without installing extra libraries.

8.

a. Which piece of code relates to which type of programming language below?

- i. High-level language - Code 1
- ii. Machine code - Code 2

Code 1	Code 2
score = score + 1	0101 0011 1010 0001 0000 0001 0011 0011 1010

b. An interpreter would translate the code between which two types of language?

<p><u>From</u> high-level language ...</p> <p>... <u>to</u> machine code</p>
--

9. A program is needed to design 3D models that can be used by pupils in schools. The programmer can use either low level language or a high-level language to implement this program. The programmer also needs to choose a type of translator to use. Give the consequences for each choice and advise the programmer on what choices should be made. [5 marks]

The following is indicative of possible factors/evidence that candidates may refer to but is not prescriptive or exhaustive:

Indicative content:

Low level language

- Harder/more time consuming to write
- Produces code that could be more efficient
- Harder to maintain/update
- Can't use libraries to speed up programming
- Likely to run very quickly
- Machine specific – would need to be re-written for a different platform

High-level language

- More use friendly to write
- Easier to maintain/update
- Can use libraries to simplify some sections
- Could be platform independent (depending on specific language chosen)
- May not run as quickly/code bloat

Translators

- Compiler will mean no access if given to the source code, making it harder for others to copy or edit the code.
- Interpreted code can be seen and edited by the end-user – provides options for open-source collaboration.
- Interpreted code needs to be translated every time the program runs, slowing down the experience for the end user.
- During development, a compiled language will highlight all identified errors, interpreted languages will only show the first error and then stop – so a compiler may help to speed up debugging.

Conclusion

- Must be a clear decision. Any decision would be appropriate if supported by the argument leading up to that point.

Accept any other reasonable comments.

10. Victoria creates a program using an Integrated Development Environment (IDE).

Describe two tools or facilities that an IDE commonly provides [4 marks].

Answer: 1 mark per bullet, max 4.

e.g.

Editor

...to enable program code to be entered/edited

Error diagnostics / debugging

...to display information about errors (syntax / run-time) / location of errors

... suggest solutions

Run-time environment

...to enable the program to be run

... check for run time errors / test the program

Translator / compiler / interpreter

...to convert the high level code into machine code / low level code / binary

...to enable code to be executed / run

Breakpoints

...to stop/pause program execution at a specific point

Watch window

...to check contents of variables

Stepping

...to execute program line by line

Syntax completion...

...suggests/corrects code

Keyword highlighting / colour coding keywords / pretty printing...

...colours command words / variables

Marks: 4

Guidance: One mark for identifying, one mark for describing. Accept description of a tool without (or with incorrect) naming of the tool.

Allow sensible descriptions which go across pairs or name other tools sensibly (e.g. editor / highlighting syntax)

Allow any sensible tool that an IDE provides (e.g. auto documentation, help tools, pretty printing)

OCR Resources: *the small print*

OCR's resources are provided to support the delivery of OCR qualifications, but in no way constitute an endorsed teaching method that is required by the Board, and the decision to use them lies with the individual teacher. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources.

Our documents are updated over time. Whilst every effort is made to check all documents, there may be contradictions between published support and the specification, therefore please use the information on the latest specification at all times. Where changes are made to specifications these will be indicated within the document, there will be a new version number indicated, and a summary of the changes. If you do notice a discrepancy between the specification and a resource please contact us at: resources.feedback@ocr.org.uk.

© OCR 2020 - This resource may be freely copied and distributed, as long as the OCR logo and this message remain intact and OCR is acknowledged as the originator of this work. OCR acknowledges the use of the following content: n/a

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications: resources.feedback@ocr.org.uk