End of Unit Quiz – Unit 2.5 Translators and facilities of languages

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

2.

a. What is meant by the term 'machine code'?

b. What is the name of **one** suitable translator that you can use?

c. What are **two** advantages for writing computer programs using a high level language?

3.

a. There is a programme written using assembly language. What is the name of the translator that is needed before this program can be executed?

b. What is **one** similarity and **one** difference between assembly language and machine code instructions?

© OCR 2017

 © OCR 2017

4.

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

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

5. What is the purpose of an assembler?

6.

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

b. For **one** of the features you have stated, explain why this feature is helpful.

7.

a. Computer programs can be written in high level languages, assembly language or machine code. What are two differences between high level languages and machine code?

b. What is the relationship between assembly language and machine code?

8.

a. What is meant by ‘code completion’ and how does this helps programmers?

b. What is meant by ‘debugging tools’ and how does this help programmers?

c. What is the name and explain the purpose of one other feature from an IDE?

9.

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

i. High Level Language

ii. Assembly Language

iii. Machine Code

Code 1	Code 2	Code 3

`LDA score`
`ADD one`
`STA score`
	`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?

c. What are some of the advantages and disadvantages of using an interpreter rather than a compiler?

10. A programmer wants to write a computer program for creating 3D models that can be used by pupils in schools. They are not sure whether to write it in assembly language or a high level language. They are not sure what kind of translator to use either. What are the consequences for each choice and advise the programmer on what to do.

Answers

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

Any suitable language, for example:

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

2.

a. What is meant by the term ‘machine code’?

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

b. What is the name of **one** suitable translator that you can use?

Compiler**.** Interpreter.

c. What are **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. There is a programme written using assembly language. What is the name of the translator that is needed before this program can be executed?

Assembler.

b. What is **one** similarity and **one** difference between assembly language and machine code instructions?

Each machine code instruction maps to one assembly language instruction

Same number of lines
Made up of opcodes and operands
Both are machine specific/

Machine code is binary / assembly language uses text**.** Machine code can be directly processed / assembly language must be translated/

4.

a. Some languages are translated using a compiler. What is the name of 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.

5. What is the purpose of an assembler?

Translate …

… assembly language code …
… into machine code (binary) …
… so that it can be processed (by the CPU)

6.

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 stated, explain why this feature is 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

7.

a. Computer programs can be written in high level languages, assembly language or machine code. What are two differences between high level languages and machine code?

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).

b. What is the relationship between assembly language and machine code?

One machine code instruction maps to one assembly language instruction
Both consist of an opcode (operation) and an operand (data/address)
Assembly code needs to be translated into machine code to be executed…
…which is done with an assembler
Both are machine specific
Both allow for low level programming…
…which can make for smaller / more efficient programs than HLL.

8.

a. What is meant by ‘code completion’ and how does this helps programmers?

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

b. What is meant by 'debugging tools' and how does this help programmers?

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

c. What is the name and explain the 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.

9.

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

i. High Level Language – Code 2

ii. Assembly Language - Code 1

iii. Machine Code - Code 3

Code 1	Code 2	Code 3

`LDA score`
`ADD one`
`STA score`
	`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?

From high level language …   … to machine code

c. What are some of the advantages and disadvantages of using an interpreter rather than a compiler?

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.

10. A programmer wants to write a computer program for creating 3D models that can be used by pupils in schools. They are not sure whether to write it in assembly language or a high level language. They are not sure what kind of translator to use either. What are the consequences for each choice and advise the programmer on what to do.

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

Indicative Content:

Assembly Language

- Harder / more time consuming to write
- Produces more lines of code
- 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 keep source code secret, 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.

This formative assessment resource has been produced as part of our free GCSE teaching and learning support package. All the GCSE teaching and learning resources, including delivery guides, topic exploration packs, lesson elements and more are available on the qualification webpages.

If you are looking for examination practice materials, you can find Sample Assessment Materials (SAMs) on the qualification webpage: Computer Science (9-1)

We'd like to know your view on the resources we produce. By clicking on 'Like' or 'Dislike' you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

Whether you already offer OCR qualifications, are new to OCR, or are considering switching from your current provider/awarding organisation, you can request more information by completing the Expression of Interest form which can be found here: www.ocr.org.uk/expression-of-interest

Looking for a resource? There is now a quick and easy search tool to help find free resources for your qualification: www.ocr.org.uk/i-want-to/find-resources/

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.

© OCR 2017 - 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