

EDULiTO

Translators and Facilities of Languages

Topic Tests



Photocopiable Resources

Terms and Conditions of Use

Your school has permission to copy this resource as many times as you require and to use it as you wish within your school/organisation.

You do not have permission to distribute it as a paper or electronic document to other schools or organisations.

Any questions? Email: edulitolearn@gmail.com

© 2016 Edulito and its licensors. All rights reserved.

Topic Test - Translators and facilities of languages

1(a) List three things that programming languages have the ability to do. [3]

.....

.....

.....

.....

.....

.....

1(b) What is the difference between a high-level and a low-level programming languages? [6]

High-Level	Low-Level

1(c) Produce a list of THREE high level programming languages. [3]

1

2

3

2 In computer science, what is the purpose of a translator? [3]

.....

.....

.....

.....

.....

3(a) What is an assembly language? [1]

.....

.....

.....

3(b) Explain the purpose of an assembler. [2]

.....

.....

.....

.....

4 Complete this table in relation to the characteristics of a compiler and an interpreter. [10]

	Compiler	Interpreter
Function		

Error checking		
Speed		
Examples		
Use		

5 (a) What is an Integrated Development Environment (IDE)? [1]

.....
.....
.....

5(b) (i) What is the role played by an **editor** in an IDE? [2]

.....
.....
.....
.....

5(b) (ii) The **editor** has features that assist with the writing and editing of code. Explain these features below. [3]

Feature	Explanation
Auto-completion	
Bracket matching	

Syntax checks	
---------------	--

5(c) What is a **run-time environment**? Why is this important? [2]

.....

.....

.....

.....

.....

Topic Test Translators and facilities of languages - Mark Scheme																						
Question Number	Answer		Additional Guidance	Mark																		
1 a	input data [1] output data [1] process calculations [1] process decisions based on certain conditions being met [1] process repetition [1] Max 3 marks			3																		
1 b	<table border="1"> <thead> <tr> <th>High-Level</th> <th>Low-Level</th> </tr> </thead> <tbody> <tr> <td>Easier to learn</td> <td>Difficult to learn</td> </tr> <tr> <td>Similar to human language</td> <td>Not similar to human language</td> </tr> <tr> <td>Slow to execute</td> <td>Fast to execute</td> </tr> <tr> <td>Easy to change</td> <td>Difficult to change</td> </tr> <tr> <td>To write a program you do not need knowledge of hardware</td> <td>Programs include link to hardware</td> </tr> <tr> <td>Used to write applications</td> <td>Used to write hardware programs</td> </tr> </tbody> </table>	High-Level	Low-Level	Easier to learn	Difficult to learn	Similar to human language	Not similar to human language	Slow to execute	Fast to execute	Easy to change	Difficult to change	To write a program you do not need knowledge of hardware	Programs include link to hardware	Used to write applications	Used to write hardware programs		Include 3 answers from each list	6				
High-Level	Low-Level																					
Easier to learn	Difficult to learn																					
Similar to human language	Not similar to human language																					
Slow to execute	Fast to execute																					
Easy to change	Difficult to change																					
To write a program you do not need knowledge of hardware	Programs include link to hardware																					
Used to write applications	Used to write hardware programs																					
1 c	Java, JavaScript, C++, Ruby, BASIC, Python etc 1 mark for each Max 3		Accept other high-level languages that have not been listed.	3																		
2	They convert/translate high-level code [1] into machine code [1] They maintain the function/logic of the original code. [1]			3																		
3 a	Assembly language is a low-level language that relates to the operation of the CPU. [1] An assembler translates assembly language [1] into machine code (a pattern of bits that the computer's processor can use to perform its basic operations).[1]			1																		
3 b	An assembler translates assembly language [1] into machine code (a pattern of bits that the computer's processor can use to perform its basic operations).[1]			2																		
4	<table border="1"> <thead> <tr> <th></th> <th>Compiler</th> <th>Interpreter</th> </tr> </thead> <tbody> <tr> <td>Function</td> <td>A compiler translates the whole program into machine code before the program is run.</td> <td>An interpreter translates code into machine code, instruction by instruction - the CPU executes each instruction before the interpreter moves on to translate the next instruction.</td> </tr> <tr> <td>Error checking</td> <td>Difficult to test individual lines of compiled code as all bugs are reported after the program has been compiled.</td> <td>Interpreted code will show an error as soon as it hits a problem, so it is easier to debug than compiled code.</td> </tr> <tr> <td>Speed</td> <td>Compilation is slow but machine code can be executed quickly.</td> <td>Interpreted code is slower to execute than compiled code.</td> </tr> <tr> <td>Examples</td> <td>Java and C++</td> <td>JavaScript, PHP, Python and Ruby.</td> </tr> <tr> <td>Use</td> <td>Used where high speed and performance are crucial</td> <td>Used for dynamic web applications.</td> </tr> </tbody> </table>		Compiler	Interpreter	Function	A compiler translates the whole program into machine code before the program is run.	An interpreter translates code into machine code, instruction by instruction - the CPU executes each instruction before the interpreter moves on to translate the next instruction.	Error checking	Difficult to test individual lines of compiled code as all bugs are reported after the program has been compiled.	Interpreted code will show an error as soon as it hits a problem, so it is easier to debug than compiled code.	Speed	Compilation is slow but machine code can be executed quickly.	Interpreted code is slower to execute than compiled code.	Examples	Java and C++	JavaScript, PHP, Python and Ruby.	Use	Used where high speed and performance are crucial	Used for dynamic web applications.		1 mark for each correct response.	10
	Compiler	Interpreter																				
Function	A compiler translates the whole program into machine code before the program is run.	An interpreter translates code into machine code, instruction by instruction - the CPU executes each instruction before the interpreter moves on to translate the next instruction.																				
Error checking	Difficult to test individual lines of compiled code as all bugs are reported after the program has been compiled.	Interpreted code will show an error as soon as it hits a problem, so it is easier to debug than compiled code.																				
Speed	Compilation is slow but machine code can be executed quickly.	Interpreted code is slower to execute than compiled code.																				
Examples	Java and C++	JavaScript, PHP, Python and Ruby.																				
Use	Used where high speed and performance are crucial	Used for dynamic web applications.																				

GCSE Computer Science (9-1) –Translators and Facilities of Languages - Topic Test

5 a	An integrated development environment (IDE) is an application used to create and develop software. [1]		1
5 b i	The IDE has a text edit area [1] that allows developers to write, edit and save a document of code. [1]		2
5 b ii	Feature	Explanation	1 mark for each correct response.
	Auto-completion	As you start to type the first part of say a procedure, it suggests or completes the procedure.	
	Bracket matching	If you forget to close a bracket while writing, coloured highlighting may help you to detect missing brackets.	
	Syntax checks	This recognises incorrect use of syntax and highlights any errors.	
5 c	A runtime environment allows you to execute the program one step at a time. [1] This is useful to test that the code is working line by line before creating the final complete program.[1]		2 /36