

2.5

**PROGRAMMING
LANGUAGES AND
IDES**

TOPIC WISE EXAM QUESTIONS

ANSWERS

GCSE

OCR

1	(a)	<p>1 mark per row</p> <table border="1" data-bbox="309 349 887 712"> <thead> <tr> <th>Statement</th> <th>Low-level</th> <th>High-level</th> </tr> </thead> <tbody> <tr> <td>The same language can be used on computers that use different hardware</td> <td></td> <td>✓</td> </tr> <tr> <td>It allows the user to directly manipulate memory</td> <td>✓</td> <td></td> </tr> <tr> <td>It allows the user to write English-like words</td> <td></td> <td>✓</td> </tr> <tr> <td>It always needs to be translated into object code or machine code</td> <td></td> <td>✓</td> </tr> </tbody> </table>	Statement	Low-level	High-level	The same language can be used on computers that use different hardware		✓	It allows the user to directly manipulate memory	✓		It allows the user to write English-like words		✓	It always needs to be translated into object code or machine code		✓	4 (AO1 1b)	<p>No mark if more than 1 tick for that row.</p> <p>Allow other indications of choice (e.g. cross) as long as clear.</p>
Statement	Low-level	High-level																	
The same language can be used on computers that use different hardware		✓																	
It allows the user to directly manipulate memory	✓																		
It allows the user to write English-like words		✓																	
It always needs to be translated into object code or machine code		✓																	
5	(a)	<p>(iii) 1 mark for feature 1 mark for matching description e.g.</p> <ul style="list-style-type: none"> • Translator / compiler / interpreter ... • ... convert to low-level/machine code • ...allow program to be executed / run • ...produce executable file (only for compiler) • ...stops execution when error found (interpreter only) • Run-time environment / output window... • ...allows program / code to be run / executed • ...shows output of the program / code • Error reporting / diagnostics • ... identify location/detail of errors • ...suggests fixes • Debugger ... • ...find errors • Stepping ... • ... execute/run the program line by line • Variable watch... 	4 (AO2 1b)	<p>Allow other sensible names for features.</p> <p>Description must add more than is given in the identification of the feature to be awarded. For example, "keyword highlighting, highlights keywords" is 1 mark for the feature only.</p> <p>If compiler and interpreter given as two distinct features, allow both (with suitable descriptions). Do not allow translator and compiler/interpreter.</p> <p>Description must match feature.</p> <p>"finding errors" is NE for description of error reporting.</p> <p>Allow sensible references to AI where appropriate. Sensible description of use needed.</p> <p>Allow other sensible features of an IDE (e.g. line numbering, auto indent, collapsed blocks, etc) with suitable description.</p> <p>For text editor / error diagnostics / debugger, allow other sensible features listed as features in the mark scheme as description (e.g. "text editor,</p>															
		<ul style="list-style-type: none"> • ... see the contents/data held in variables • Break points ... • ... will allow the program to stop at a chosen / set position • Text/code editor... • ...allows program code to be written / entered / changed • ...allows errors to be fixed • Pretty printing // keyword highlighting... • ... allows keywords / variables to be coloured / identified • Keyword completion // syntax suggestion... • ...suggests code/syntax when first part entered. 		<p>provides pretty printing", "debugger, provides stepping")</p>															

2022

4	(b)	(ii)	<ul style="list-style-type: none"> high-level stops // crashes no executable without 	5 (AO1 1b, AO2 1b)	Ignore spelling errors.
---	-----	------	---	--------------------------	-------------------------

SAMPLE

5	a	<ul style="list-style-type: none"> To convert it to binary/machine code The processor can only understand machine code 	1 (AO1 1a)	Maximum 1 mark
	b	<ul style="list-style-type: none"> Compiler translates all the code in one go... ...whereas an interpreter translates one line at a time Compiler creates an executable... ...whereas an interpreter does not/executes one line at a time Compiler reports errors at the end... ...whereas an interpreter stops when it finds an error 	4 (AO1 1b)	1 mark to be awarded for the correct identification and one for a valid description up to a maximum of 4 marks. No more than 2 marks for answers relating only to interpreters and no more than 2 marks for answers only relating to compilers.
	e	<ul style="list-style-type: none"> Error diagnostics (any example) Run-time environment Editor (any feature such as auto-correct, auto-indent) Translator Version control Break point Stepping 	2 (AO1 1a)	1 mark per bullet to a maximum of 2 marks. Only 1 example per bullet, e.g. auto-correct and auto-indent would only gain 1 mark.

2021

	(b)	<ul style="list-style-type: none"> Transistor has two states 1 represents on, 0 represents off Each transistor stores one bit Multiple transistors used to store a binary value 	2	Allow values for BP1
--	-----	--	---	----------------------

2020

2	(b)	<p>1 mark per bullet to max 2</p> <ul style="list-style-type: none"> • Easier/quicker for humans to write • Easier/quicker to read / understand / remember • Easier/quicker to maintain / debug / spot errors • ...because code is closer to English / uses English words • Less code to write • ...because one HLL instruction represents many assembly instructions • Portable (between processors) // will work with different types of computer 	2 AO1 1b(2)	<p>Accept "human language" as English for BP4</p> <p>"Easier to use" is too vague.</p>
2	(c)	<p>1 mark per bullet to max 2</p> <ul style="list-style-type: none"> • Each character (in character set) has a unique (binary) number/value • Each character in the string is assigned its associated number/value • The (binary) value of each character is stored/combined (in order) • ... by example e.g. The binary value for D, then for r, then for u • Uses ASCII/Extended ASCII/Unicode 	2 AO2 1a(2)	

2019

2	(c)	<p>1 mark per bullet to max 4, 2 mark max per method</p> <ul style="list-style-type: none"> • Compiler • ...translates code in one go / all at once • ...produces an executable file // does not need to be compiled again • Interpreter • ...translates code line by line. • ...will be interpreted / translated every time it is run. 	4 AO1 1b (4)	Mark first method only in each section
---	-----	--	-----------------	--

7	(a)	(ii)	<p>1 mark per bullet, max 2.</p> <ul style="list-style-type: none"> aimed at humans//understandable by humans / programmers English like structure / syntax Must be translated/compiled/interpreted (before it can be run) Allows programmer to deal with the problem instead of considering the underlying hardware // an abstraction from the hardware // hardware independent // portable 	<p>2</p> <p>Allow examples of keywords (eg IF / ELSE / WHILE) as 2nd bullet point.</p> <p>Do not award marks for naming languages such as Java , Python, etc.</p> <p>Do not award marks for stating what a high level language isn't (i.e. describing what low level code is).</p> <p>Do not allow "easy to use"</p> <p>Do not allow 'has to be converted' without into what i.e machine code etc.</p>
7	(b)		<p>1 mark per bullet, max 4.</p> <p>e.g.</p> <ul style="list-style-type: none"> Editor <ul style="list-style-type: none"> ...to enable program code to be entered/edited Error diagnostics / debugging <ul style="list-style-type: none"> ...to display information about errors (syntax / run-time) / location of errors ... suggest solutions Run-time environment <ul style="list-style-type: none"> ...to enable to the program to be run ... check for run time errors / test the program Translator / compiler / interpreter <ul style="list-style-type: none"> ...to convert the high level code into <u>machine code</u> / <u>low level code</u> / <u>binary</u> ...to enable to code to be executed / run 	<p>4</p> <p>One mark for identifying, one mark for describing. Accept description of a tool without (or with incorrect) naming of the tool.</p> <p>Allow sensible descriptions which go across pairs or name other tools sensibly (e.g. editor / highlighting syntax)</p> <p>Allow any sensible tool that an IDE provides (e.g. auto documentation, help tools, pretty printing etc.)</p>

4	f	<p>1 mark for identification, 1 for matching description e.g.</p> <ul style="list-style-type: none"> • Error diagnostics/debugger • ...highlight errors/suggest changes • Run-time environment • ...Lets you run/test the program • Text editor • ...highlight key words • ...auto-indent • ...to type/edit source code • ...Auto-complete • ...highlight syntax errors • Versioning tools • ...To allow for tracing back • ...To create new files with changes • Stepping/breakpoints • ...Allow tracing of algorithms 	<p>4</p> <p>Do not allow auto-documentation. Can get description mark, without identification/incorrect identification</p> <p>Allow:</p> <ul style="list-style-type: none"> • Variable watch/window • See how the values change <p>Do not allow compiler/interpreter</p>
4	g	<p>Max 2 for compiler, 2 for interpreter</p> <p>Compiler</p> <ul style="list-style-type: none"> • To convert to low-level in one go • Create an executable//export the file • To distribute the software • Users will have no access to source code... • ...so no-one can edit/steal/copy the code/program • Use for error detection <p>Interpreter</p> <ul style="list-style-type: none"> • To convert to low-level <u>line by line</u> • To test the program // to find errors • stops running when it finds an error//shows the location of the error when found • it is quicker (compared to compiler) to re-interpret than re-compile 	<p>4</p> <p>The uses must be different for compiler and interpreter</p>

2015

5	a	i	<p>High level code :</p> <ul style="list-style-type: none"> • human oriented code / written by programmers • contains words for commands / closer to English/natural language • Machine independent /Portable to different systems • Needs to be translated before it can be executed. • Problem based • One (high level) command equates to many machine code instructions. <p>Machine code:</p> <ul style="list-style-type: none"> • Code for the CPU to execute / not readily understandable by humans • binary instructions • specific to a particular (type of) computer / not portable to different systems • does not need to be translated <p>[max 2 marks for each type of code]</p>	4	<p>Award marks for correct points about machine code made under high level code and vice versa.</p> <p>Do not accept Machine code is in Hex</p>
		ii	<ul style="list-style-type: none"> • To translate the <u>high level code into machine code</u> • To pick up (syntax) errors 	1	<p>Translate to object code is acceptable</p> <p>Accept "errors" on its own, but do not accept answers referring specifically to logic or runtime errors.</p>

**If you found this
useful, drop a follow
to help me out!**

THANK YOU!

GCST