AQA Computer Science A-Level

4.6.3 Types of program translator

Past Paper Mark Schemes

January 2010 Comp 2

4	(a)		So that source code cannot be accessed by users; Users do not need to have an interpreter/compiler/translator//users do not need programming environment; Users do not need knowledge of the programming environment; So that the program will execute more quickly; **NE** it's faster **NE** does not need to be compiled each time executed/run **R** saves disk space **MAX 2**	2

4	(b)		Can't know what type of processor will be in user's computer//Internet users have range of computers/devices with different processors; A compiled program will only execute on a processor of specific type/family/with same instruction set//A program run using an interpreter can execute on a computer with any type of processor; **A** References to just different types of computer/device rather than specifically processors **NB** Virtual Machine **R** No compiler exists **R** computers may have different web browsers/software	2

January 2011 Comp 2

7	b		Compiler produces object code to distribute that is difficult to reverse engineer/ no need to distribute the source code; Compiler optimises the code // The object code /program runs faster (as it does not need translating); **NE** "Runs faster", if not clear whether this applies to the program or the compiler. The target computer has no need to have the original compiler; Object code can be installed on target computer; No interpreter available for target machine;	2

January 2012 Comp 2

1	**d**		A compiler produces object code/machine code; whilst an interpreter does not produce any object code; Interpreted code will execute slower; than executing the object code produced by a compiler; You always need the interpreter to interpret source code; but you do not need the compiler to execute a compiled program; Once compiled source code is no longer required to run the program; An interpreter always needs source code at runtime; Compiled code can only be executed on a machine with the same processor type / instruction set; Interpreted code is more portable; A compiler translates the whole source code (at once); An interpreter analyses the code line by line; **NE** – reads	**MAX 4**

June 2010 Comp 2

7	**(d)**	**(i)**	Assembler;	1

June 2013 Comp 2

4	**(b)**		Analyses statement by statement each line of source code; **A.** runs/translates/executes line by line **R.** compiles (line by line) Calls routines to carry out each instruction/statement	**MAX 2**

June 2016 AS Paper 2

09	3	**Marks are for AO1 (understanding)** A compiler produces object/machine code (**A.** executable file) whilst an interpreter does not // once code has been compiled it does not (normally) need to be recompiled whilst an interpreter has to translate code every time a program is run // if using an interpreter it needs the source code each time it executes the program whereas a compiler only needs to use the source code once; A compiler translates the whole of the source code into object code (prior to execution) whilst an interpreter translates and executes line by line; The object code produced by a compiler will execute faster once it is compiled than interpreting the source code (every time the program is run); An interpreter can run (syntactically correct) parts of a program whilst there are syntax errors in other parts of it, which a compiler cannot; **Max 2**	2

June 2017 AS Paper 2

04	1	**4 marks for AO1 (understanding** A compiler produces object code/machine code/executable file; An interpreter does not produce any object code; A compiler translates the whole source code (at once); An interpreter analyses the code line by line; **A.** Deals with, translates, processes, **R.** Runs through, reads, convert A compiler will not produce an executable file if an error is encountered; An interpreter will run the program up until the first error; Interpreted code will execute slower than executing the object code produced by a compiler; **A.** opposite You do not need the compiler to execute a compiled program; When running interpreted code, the interpreter always needs to be present Once compiled source code is no longer required to run the program; An interpreter always needs source code at runtime; Compiled code can only be executed on a machine with the same processor type / instruction set; Interpreted code is more portable; **Max 3** if all points made about either interpreter or compiler.	**4**

June 2009 Comp 2

2	(a)	Compiler **R** Interpreter **A** Misspellings where meaning remains clear e.g. complier **R** More than one answer e.g. compiler or interpreter	**1**
	(b)	Assembler **A** Misspellings where meaning remains clear **R** More than one answer	**1**

Specimen AS Paper 2

10	**1**		**Mark is for AO1 (understanding)** Version: B;	1

10	**2**		**Marks are for AO1 (understanding)** A compiler produces object code whilst an interpreter does not; A compiler translates the whole of the source code into object code whilst an interpreter translates line by line; The object code produced by a compiler will execute faster, (once it is compiled) than interpreting the source code (every time the program is run) An interpreter can run (syntactically correct) parts of a program whilst there are syntax errors in other parts of it, which a compiler cannot; **Max 2**	MAX 2

10	**3**		**Marks are for AO1 (understanding)** Intermediate code is not (directly) executable // Intermediate code will by run/interpreted by a virtual machine // Compiled into an executable just before running/just in time; Intermediate code can be run on different computing platforms // One solution can be targeted at multiple platforms; **Max 2**	2

Specimen Paper 2

06	**5**	**All marks AO1 (understanding)** So that source code cannot be accessed by users; So that it is more convenient for users to run it // users do not need to have an interpreter; So that the program will execute more quickly; **Max 2**	2

06	**6**	**All marks AO1 (understanding)** **1 mark:** Can't know what type of processor will be in user's computer//Internet users have range of computers/devices with different processors; **A.** References to just different types of computer/device rather than specifically processors **1 mark:** A compiled program will only execute on a processor of specific type/family/with same instruction set//A program run using an interpreter can execute on a computer with any type of processor; **R.** No compiler exists	2