Chapter 20: System software: Answers to coursebook questions

3 **a** **i** Analysis (1) and synthesis or object code creation (1).

ii 1 mark each for any of the following up to a maximum of 2: Lexical (1), syntax (1) or semantic (1) analysis.

iii 1 mark each for any of the following up to a maximum of 2: Evaluation of expressions (1), optimisation (1), machine code or object code generation (1).

b <Digit>::= <0|1|2|3|4|5|6|7|8|9> (1)

<Sign>::= <+|-> (1)

<Unsigned integer>::= <Digit>|<Digit><Unsigned integer> (1)

<Signed integer>::= <Sign><Unsigned integer> (1)

Note that this is an answer to the question as stated but it would allow an integer to start with a zero. The set of definitions would need to be modified to ensure that an integer could not start with a zero.

c a 6 + b c / + (2)

d This converts to a + (3*b)–(6*c) (2)

4 This is Question 1 in 9608 Paper 31 June 2015. At the time of writing the published mark scheme is available on the Cambridge International School Support Hub (requires registration). The Examiners Report for the June 2015 series is also available there and this may contain comments specific to this question.

The following are what the author of this chapter in the Teacher Resource would suggest as reasonable answers with alternatives suggested where appropriate. Where a suggested answer includes bullet points, each bullet point would be worth one mark up to the maximum mark allocation for the question.

a **i** The semi-colon is missing

ii 2 is not a valid variable

iii e is not a valid letter

b <assignment statement> ::= <variable> = <variable><operator><variable>

<variable> ::= <letter>|<letter><letter> |<letter><letter><letter>

<letter> ::= a|b|c|d

<operator> :: =+|-|*|÷

c There are two options here:

<letter> | <letter><variable> OR <letter> | <variable><letter>

d **i** Saying debugging is easy or fast is not sufficient (saying easier or faster would be better). The best alternative is to state that incomplete code can be debugged line by line.

ii There are two slightly different reasons here:

- When a compiler has produced executable code this will run without the source code being present

- It is very difficult to re-create the source code from the executable code (by what is described as reverse-engineering).

Cambridge International AS & A Level Computer Science 9608 paper 31 Q1 June 2015

5 This is Question 6 in 9608 Paper 31 November 2015. At the time of writing the published mark scheme is available on the Cambridge International School Support Hub (requires registration). The Examiners Report for the November 2015 series is also available there and this may contain comments specific to this question.

The following are what the author of this chapter in the Teacher Resource would suggest as reasonable answers with alternatives suggested where appropriate. Where a suggested answer includes bullet points, each bullet point would be worth one mark up to the maximum mark allocation for the question.

a The first part of the answer is the statement that a process refers to a program that has been loaded into memory and the execution of the program has begun. There are a number of possible statements about a program. The program could be the source code or the executable code. In the context of this question it is best to define the program as the code that is not being executed.

b **Running to ready:** In the running state the process is being executed. The scheduler might return it to the ready state for reasons associated with the scheduling algorithm being used. One example is that the allocated time slice has come to an end. The process could have continued and this still applies when it has been returned to the ready state.

Ready to running: In the ready state the process is capable of being executed. The scheduler returns it to the running state as and when the scheduling algorithm dictates that it is the turn for the process.

Running to blocked: This time the running process reaches a stage where it cannot continue processing without some delay. The usual reason is that the process needs some I/O processing, which is very slow compared to processing only requiring memory access. An interrupt halts the process. When the I/O processing has completed the blocked process can return to the ready state.

c The scheduler controls which process should next enter the running state. This decision is based on a choice between the processes that are in the ready state. Therefore when a process in the blocked state is ready to continue processing once more it must be moved to the ready state to await its turn.

d There may be many programs stored in the backing store that are ready for processing. The high-level scheduler makes decisions about which one should be moved into memory to join the ready queue.

Cambridge International AS & A Level Computer Science 9608 paper 31 Q6 November 2015