

Revision Sheet

A-Level Computer Science · Unit 16 System Software

12 vocabulary items · 24 marks of exam questions (4 questions)

Part 1 · Vocabulary review answer key. Section A wants the exact English spelling.

Question	Answer	Marks
1	grammar	1
2	process control block	1
3	inter-process communication	1
4	shared memory	1
5	virtual address space	1
6	segmentation	1
7	syntax analysis (parsing)	1
8	abstract syntax tree	1
9	Backus-Naur Form	1
10	syntax diagram	1
11	infix	1
12	Reverse Polish Notation	1

13	<p>One mark for each correct line connecting one stage of compilation to a description</p> <table> <tr> <th>Stage of compilation</th><th>Description</th></tr> <tr> <td>Lexical analysis</td><td>minimising a program's execution time and memory requirement</td></tr> <tr> <td>Syntax analysis</td><td>converting an intermediate representation of source code into an executable form</td></tr> <tr> <td>Code generation</td><td>converting a sequence of characters into a sequence of tokens</td></tr> <tr> <td>Optimisation</td><td>directly executing instructions written in a scripting language</td></tr> <tr> <td></td><td>using parsing algorithms to interpret the meaning of a sequence of tokens</td></tr> </table>	Stage of compilation	Description	Lexical analysis	minimising a program's execution time and memory requirement	Syntax analysis	converting an intermediate representation of source code into an executable form	Code generation	converting a sequence of characters into a sequence of tokens	Optimisation	directly executing instructions written in a scripting language		using parsing algorithms to interpret the meaning of a sequence of tokens
Stage of compilation	Description												
Lexical analysis	minimising a program's execution time and memory requirement												
Syntax analysis	converting an intermediate representation of source code into an executable form												
Code generation	converting a sequence of characters into a sequence of tokens												
Optimisation	directly executing instructions written in a scripting language												
	using parsing algorithms to interpret the meaning of a sequence of tokens												

14(a)	<p>One mark per point</p> <ul style="list-style-type: none"> Code generation Optimisation
14(b)	<p>One mark per point (Max 2)</p> <ul style="list-style-type: none"> It checks that the code matches the grammar of the language // It checks that the tokens conform with the rules of the programming language Syntax errors are reported A parse tree is produced.

15(a)	<p>One mark for each marking point (Max 2)</p> <ul style="list-style-type: none"> • <code><character>::=</code> • <code>\$ % & * #</code> <p>Complete answer <code><character>::= \$ % & * #</code></p>
15(b)(i)	For example: \$A9E3
15(b)(ii)	<p>One mark for each marking point (Max 4)</p> <ul style="list-style-type: none"> • <code><password>::=<character> ...</code> • <code>... <code></code> • <code><code>::= ...</code> • <code>... <digit> <capital_letter></code> • <code>... <digit><code> <capital_letter><code></code> <p>Complete answer <code><password>::=<character><code></code> <code><code>::=<digit> <capital_letter> <digit><code> <capital_letter><code></code></p>

16(a)	<p>One mark for each correct line from Operating System Term to Description</p> <table border="1"> <thead> <tr> <th>OS term</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Multi-tasking</td><td>Using secondary storage to simulate additional main memory</td></tr> <tr> <td>Paging</td><td>Managing the processes running on the CPU</td></tr> <tr> <td>Interrupt handling</td><td>Managing the execution of many programs that appear to run at the same time</td></tr> <tr> <td>Scheduling</td><td>Locating non-contiguous blocks of data and relocating them</td></tr> <tr> <td>Virtual memory</td><td>Transferring control to another routine when a service is required</td></tr> <tr> <td></td><td>Reading/writing same-size blocks of data from/to secondary storage when required</td></tr> </tbody> </table>	OS term	Description	Multi-tasking	Using secondary storage to simulate additional main memory	Paging	Managing the processes running on the CPU	Interrupt handling	Managing the execution of many programs that appear to run at the same time	Scheduling	Locating non-contiguous blocks of data and relocating them	Virtual memory	Transferring control to another routine when a service is required		Reading/writing same-size blocks of data from/to secondary storage when required
OS term	Description														
Multi-tasking	Using secondary storage to simulate additional main memory														
Paging	Managing the processes running on the CPU														
Interrupt handling	Managing the execution of many programs that appear to run at the same time														
Scheduling	Locating non-contiguous blocks of data and relocating them														
Virtual memory	Transferring control to another routine when a service is required														
	Reading/writing same-size blocks of data from/to secondary storage when required														
16(b)	<p>One mark for each correct statement (Max 4)</p> <ul style="list-style-type: none"> • An interpreter examines source code one statement at a time • Check each statement for errors • ...If no error is found the statement is executed • ...If an error is found this is reported and the interpreter halts • Interpretation is repeated for every iteration in repeated sections of code/in loops • Interpretation has to be repeated every time the program is run 														