

5(a)(i)	1 mark per bullet point, max 2 marks • Installs <u>driver</u> software for devices connected to external ports • Manages communication between devices • Manages <u>hardware</u> interrupts
5(a)(ii)	1 mark per bullet point, max 2 marks • Prevents unauthorised access • ... by providing authentication // by validating users and processes • Implements access rights and permissions • Makes provision for recovery of lost data • Carries out Operating System security updates as available • Carries out auditing and keeps logs of activity
5(b)(i)	One mark for benefit, one mark for appropriate matching expansion e.g. • Less code needs to be developed as code is pre-written • ... thereby reducing the time to write a program • The library routine will be pre-tested and documented • ... reducing the time taken to test a program // giving confidence that the routine works as it should • The code for the library routine can be written in a different programming language • ... which allows the use of special features of that language • The user's source code is more readable • ... as the library routine is simply called in the source code rather than being written out in full • The library could contain the code for complex routines • ... thus allowing the programmer to include modules they might not be able to code
5(b)(ii)	1 mark for drawback, 1 mark for appropriate matching expansion e.g. • Compatibility issues • ... the library routine may not work with current program • Not guaranteed to be thoroughly tested • ... there could be unexpected problems (bugs/virus) • The library routine may not match needs exactly // The library routine may give unexpected results • ... so it will need to be edited and tested • If the library routine is later edited or changed (by someone else) • ... there could be unexpected errors or results
5(c)	1 mark per bullet point, max 2 marks • Prettyprint • Expanding and collapsing code blocks • <u>Automatic</u> / <u>smart</u> indentation / formatting

4(a)(i)	1 mark for each bullet point (max 2) e.g. • Secondary storage space is divided into file allocation units • Space on secondary storage is allocated to particular files • Maintains/creates directory structures • Specifies the logical method of file storage e.g. FAT or NTFS • Provides file naming conventions • Controls user access // implements access rights //implements password protection // Makes file sharing possible • Controls access to the file by other software • Specifies tasks that can be performed on a file e.g. open, close, delete, copy, create, move, etc • Allows searching for a file
4(a)(ii)	1 mark for each bullet point (max 2) • To allow data to be retrieved / restored when lost • To automatically make a duplicate copy of data... • ...so the user does not have to remember to backup data • To make regular duplicate copies of data
4(b)(i)	1 mark for each bullet point (max 3) e.g. • Less of the teacher's storage space is used • ... so more student work can be stored • Transmission time is reduced • ... so the teacher does not have to wait as long for it to arrive • Bandwidth usage is reduced • ... so other transmissions are not adversely affected • Less data allowance is used on the teacher's email system
4(b)(ii)	1 mark for each bullet point (max 3) • Run-Length Encoding // RLE • Repeated sequences of the same characters are replaced by • ... a single copy of the character • ... and a count of the number of characters
4(c)	1 mark for each bullet point (max 3) e.g. • It saves development time for the student • ... by using pre-written code • There is less testing to do // the routines are more likely to work correctly • ... because they have already been tested and are error-free • It results in a more robust program • Subroutines can be used that they do not have the skill to write • ... which enables a more complex program to be written

4(d)(i)	1 mark for correct name, 1 mark for matching description • Expand/collapse code blocks... • ... sections of source code that are part of the same block can be expanded to see the content or collapsed so that the overall code is seen • Auto-indentation // auto-formatting... • ... automatically indents/formats code as the user types so that the structure is clear // aids readability
4(d)(ii)	1 mark for correct name, 1 mark for matching description • Breakpoints ... • ... stops the code running on a set line to view the current status/variable contents/program flow • Report window // variable watch window ... • ... shows the values in variables/data structures and how they change when each line is run

6	1 mark each to max 4 Max 3 marks for each management task: <i>Memory management: Max 3 marks</i> • Stores data from all currently running programs concurrently in RAM • Stops the data from overwriting each other in RAM/primary storage • Decides which processes should be in main memory • Makes efficient use of memory <i>Process management: Max 3 marks</i> • Allows one process to be paused whilst another process can be actioned • Decides which process is to be run next • Switches between processes to allow them to share the use of the processor • Identification/description of scheduling
---	--