

Chapter 5 Student Book Answers

5.1 What you should already know

- 1 Devices such as washing machine and ovens have simple repetitive tasks which are controlled by user pushing a button or turning a dial ... therefore no need for an OS.
- 2 a) For example, Microsoft Windows 10, Apple MacOS, Google Android, IOS (mobile phones/tablets only)
 - b) **Different OS versions developed because**
 - different interfaces
 - different connectivity methods
 - different screen sizes
 - need to manage cellular and wireless connectivity as well as phone access
 - desktop OS allow Apps to continue consuming memory until the system gets very slow; mobile phones/tablets, with their limited memories, can't allow this and need better management/control from the OS
 - need to optimise for small memories and power use.
- 3 a) GUI – uses graphical interface which involves icons and selection by pointing devices or touch screen.
 CLI – a user interface is keyboard where user keys in commands to carry out actions; exact instructions need to be used.

b)

interface	advantages	disadvantages
command line interface (CLI)	<ul style="list-style-type: none"> • the user is in direct communication with the computer • the user is not restricted to a number of pre-determined options • it is possible to alter computer configuration settings 	<ul style="list-style-type: none"> • the user needs to learn a number of commands to carry out basic operations • all commands need to be typed in which takes time and can be error prone • each command must be typed in using the correct format, spelling, and so on
graphical user interface (GUI)	<ul style="list-style-type: none"> • the user doesn't need to learn any commands • it is more user-friendly; icons are used to represent applications • a pointing device (such as a mouse) is used to click on an icon to launch the application – this is simpler than typing in commands 	<ul style="list-style-type: none"> • this type of interface uses up considerably more computer memory than a CLI interface • the user is limited to the icons provided on the screen • needs an operating system, such as windows, to operate, which uses up considerable memory

c) CLI

- programmer, analyst or technician
- basically anybody who needs to have direct communication with a computer when developing new software or problem-solving, for example.

GUI

- the end-user who doesn't need any great knowledge of how a computer works
- a person who uses the computer to run software, play games or manipulate photographs, for example.

4 Examples include: paper tape and punched cards to start-up computers (other methods are accepted)

5

- data sent to printer buffer from computer
- contents of buffer sent to printer and document starts to be printed
- processor carries on with other tasks while printing continues in background
- if printer runs out of paper, runs out of ink, paper jam (etc.) then it send out an interrupt signal
- if interrupt sent out, message is displayed on computer screen requesting user to resolve issue
- once all the data from the buffer is printed, printer sends an interrupt signal to the processor requesting more data
- depending on its priority, interrupt is serviced
- once interrupt is serviced, more data is sent to the printer buffer and the above stages are repeated until all 1000 pages are printed out

Activity 5A

1 a) i) A computer needs an OS because

- it provides platform on which programs can run
- hardware devices are difficult to use without OS/hides complexity of hardware
- it acts as an interface between users and hardware.

ii) Management tasks

- process management
- interrupt handling
- file management
- memory management
- hardware management
- software platform
- security management
- user interfaces.

b) i) DLL files

- shared library files
- code saved separately from main .exe files
- code only loaded into memory as required at run time
- DLL files can be made available to several apps at same time.

ii) Drawbacks of using DLL files

- DLL files aren't integrated into programs that use them which leads to shared code.
- The programs using DLL files aren't self-sufficient therefore a program requiring these files won't work if they are not available or are incompatible with the version of the software.
- Shared files are an issue if they have been updated at a later point making them potentially incompatible later.
- Because DLL libraries are outside programs that use them, they could be modified by another programmer which may remove any software protection or even introduce viruses or spyware into programs that use them.
- DLL-dependent programs usually run slower than those that use static library routines.

2 Disk formatter – prepares HDD to allow data to be stored.

Virus checker – checks for viruses and then quarantines/removes them.

File compression – reduces file size to save storage using lossy/lossless compression algorithms.

Back-up software – makes copies of files on separate medium in case of data corruption/loss.

Firewall – prevents unauthorised access to computer from external sources.

3 a) File compression – reduces file size so they can be sent.

b) Back-up – in case files/data are lost, a copy can be reinstalled.

c) Defragmentation – removes unused sections of HDD and makes it faster to access files.

d) Anti-virus – quarantines and removes any viruses found.

e) Formatters – can repair disks if bad sectors are found.

5.2 What you should already know

1 a) Compiler and Assembler

b) Interpretation

2 a) PyCharm (example there are many more)

b) Source code editor with colour coding, automatic layout and autocompletion, alternative interpreters, a debugger with single stepping, breakpoints and a report window showing the value of variables whilst single stepping through the program.

c) Most useful for me autocompletion to save time. (Example answer)

Activity 5B

1 a) i) An assembler translates programs written in assembly low-level languages and a compiler translates programs written in a high-level language.

ii) A compiler produces an object program an interpreter doesn't

b) See 5.2.2

2 a) a suite of programs used to write and test a computer program written in a high-level programming language.

b) Source code editor used to write and edit programs. An interpreter used to run programs. Runtime environment with debugger.

End of chapter questions**1 a) Benefits of library routines**

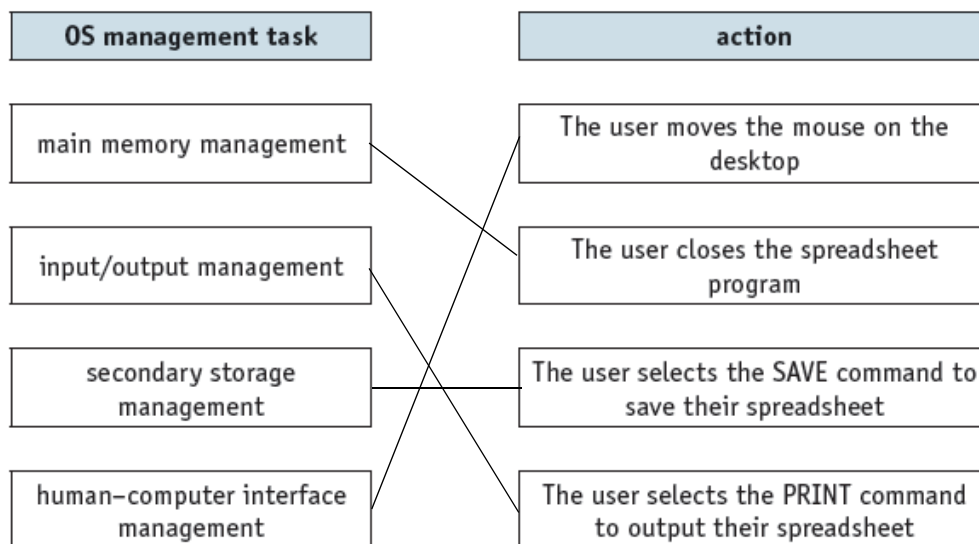
- code already written therefore saves programmer time
- code has been used by many users therefore thoroughly tested and error-free
- programmer has access to many complex routines which may be outside his knowledge if he had to write them
- code should conform to industry standards which leads to more robust/compatible programs.

b) i) Benefits of DLL files

- executable code is smaller because DLL files only loaded at run time
- changes/improvements to DLL file code is done independently of main program therefore there is no need to recompile the main program
- single DLL file can be made available to several applications at the same time
- saves on memory space

ii) Drawbacks of DLL files

- executable code is not self-contained therefore DLL files need to be included at run time
- linking software must be available at run time so that links to DLL files can be made
- DLL files must always be present otherwise inability to find the file at run time would lead to .dll errors occurring
- unexpected changes to DLL files/corrupted DLL files could lead to the main program crashing or giving unexpected results
- malicious changes to DLL files (For example, viruses) could lead to user's computer becoming infected or stored data being deleted/corrupted.

2 a)**b) i) file compression software****ii) back-up software****iii) disk repair software****iv) anti-virus software**

3 a) in case files are deleted/corrupted (accidentally or on purpose)

b)

- allows user to return computer to status before the problem occurred
- it may involve reverting back to earlier versions of the software or files
- if a file is lost or the user forgot to save it, an earlier version (from auto save) could be retrieved
- if OS updated and a newer version causes problems when running software, there may be a need to revert back to the earlier version of the OS

c) **Points to consider**

- As an HDD becomes full, blocks used for files will become scattered all over the disk surface (in potentially different sectors and tracks as well as different surfaces) ...
- ... this will happen because files will become deleted, partially deleted, extended and so on over time ...
- ... the consequence of this is
 - slower data access time
 - the HDD read-write head will now require several movements just to find and retrieve the data making up the required file
 - It would obviously be advantageous if files could be stored in contiguous sectors considerably reducing HDD head movements ...
 - ... due to the different operation of SSDs when accessing data, this is less of a problem when using solid state devices.
- A disk defragmenter will rearrange the blocks of data to store files in contiguous sectors wherever possible ...
- however, if the disk drive is almost full, defragmentation may not work.
- Defragmentation obviously allows for much faster data access and retrieval since the HDD will now require fewer read-write head movements to access and read files.
- Some defragmenters also carry out clean up operations
 - data blocks can become damaged after several read/write operations ...
 - in which case, they are flagged as “unusable” ...
 - and any subsequent write operation will avoid writing data to data blocks which have become affected.

4 **Assemblers** – are used to translate programs written in low-level languages only.

Compilers – are used to translate programs written in high-level languages only.

Interpreters – are used with programs written in high-level languages only, no object program is produced.

Both assemblers and compilers translate source code into object code. Both compilers and interpreters are used with high-level languages. Both an interpreter and a compiler can be used within an IDE.

5 Any **four** from

- Context-sensitive help
- Syntax checking (on entry)
- Automatic indentation
- Type checking (Parameter checking)
- PrettyPrinting
- Highlight structure blocks (e.g. selection, iteration)
- Highlight any undeclared variables
- Highlight any unassigned variables