In this chapter, you have learnt about:

- ✔ system software
- ✔ application software
- ✔ utility programs
- ✔ the role and the function of an operating system
- ✔ how hardware, firmware and operating systems are used to run application software
- ✔ the role and operation of interrupts
- ✔ types of programming language – high-level and low-level
- ✔ translation software – compilers, translators and assemblers
- ✔ an Integrated Development Environment (IDE).

Key terms used throughout this chapter

utility programs (utilities) – part of an operating system which carries out certain functions such as virus checking, defragmentation and screensaver

malware – programs (such as viruses, worms and Trojan horses) that are installed on a user's computer with the aim of deleting, corrupting or manipulating data illegally

anti-virus software – software that quarantines and deletes files or programs infected by a computer virus; the software can run in the background or be initiated by the user

heuristic checking – checking software for behaviour that could indicate a possible virus

quarantine – to isolate (in order to later delete) a file or program identified by anti-virus software as being infected by a virus

defragmentation – a process that reorganises sectors on an HDD by rearranging blocks of data so that they are contiguous

contiguous – next to each other

back-up – make copies of files onto another storage media in case the original file becomes corrupted or is deleted

screensaver – software that supplies a still or moving image on a monitor if a computer has been inactive for a period of time

device driver – software that communicates with the operating system and translates data into a format understood by the device

descriptor – a collection of information about a device plugged into a USB port; this can be vendor ID (VID), product ID (PID) or serial number

operating system – software that provides an environment in which applications can run and also provides an interface between computer and human operator

boot up/bootstrap loader – a small program that is used to load other programs to correctly 'start-up' a computer system

EEPROM – stands for electronically erasable programmable ROM

human computer interface (HCI) – an interface supplied by the operating system to 'hide' the complexities of the software and hardware from the human user

command line interface (CLI) – an interface which allows communication with the computer by typing in commands using a keyboard

graphical user interface (GUI) – an interface that uses icons to represent apps and tasks which the user can select/launch by clicking on a mouse or using a touch screen

windows icons menu and pointing device (WIMP) – an interface that uses a pointing device such as a mouse to select options from screen icons or a menu

post-WIMP – a modern touch screen interface system that allows actions such as pinching and rotating

memory management – the part of an operating system that controls main memory

security management – the part of an operating system that ensures the integrity, confidentiality and availability of data

hardware management – the part of an operating system that controls all input and output devices; it is made up of sub-systems such as printer management

buffer – a memory area used to store data temporarily

file management – part of an operating system that manages files in a computer (for example, the ability to create, delete, copy, open, close and rename files)

interrupt – a signal sent from a device or software to a microprocessor requesting its attention; the microprocessor suspends all operations until the interrupt has been serviced

multitasking – a function that allows a computer to process more than one task/process at a time

administrator – a person responsible for the upkeep and maintenance of a computer system that involves multi-user function

user account – an agreement that allows an individual to use a computer; the user needs a user name and password to enter the user's area

error handling routine – a routine in a program or operating system that recognises and recovers a system from abnormal inputs or hardware faults (for example, recovery from an attempt to divide by zero)

firmware – a program that provides low level control for devices

interrupt priority – the priority assigned to an interrupt are given a priority so that the microprocessor knows which one needs to be serviced first and which interrupts are to be dealt with quickly

service (an interrupt) – when an interrupt is received, some action needs to be taken by the processor depending on what caused the interrupt; until this is resolved (that is, it is serviced), the interrupt cannot be removed to allow the processor to continue

interrupt service routine (ISR) – software that handles interrupt requests (for example, when the printer out of paper) and sends a request to the CPU for processing

machine code – a binary programming language, a program written in machine code can be loaded and executed without translation

high-level language (HLL) – a programming language that is independent of computer hardware, a program written in a HLL needs to be translated into machine code before it is executed.

low-level language (LLL) – a programming language that is dependent on computer hardware, both machine code and assembly language are LLLs

assembly language – a programming language that is dependent on computer hardware, a program written in an assembly language program needs to be translated into machine code before it is executed

assembler – a computer program that translates programming code written in assembly language into machine code

compiler – a computer program that translates a source program written in a high-level language to machine code

translator – converts a program written in a high-level language program into machine code

interpreter – a computer program that analyses and executes a program written in a high-level language line by line

Integrated Development Environment (IDE) – a suite of programs used to write and test a computer program written in a high-level language

debugging – finding errors in a computer program by running or tracing the program

prettyprinting – displaying source code using different colours and formatting, which make the code easier to read and understand

report window – a separate window in the runtime environment of an IDE that shows the contents of variables during the execution of a program