

Cambridge IGCSE · Computer Science 0478 · Notes

Types of software and interrupts

Read online at pastopic.com/cambridge/computer-science-0478/notes/types-of-software-and-interrupts

Last checked 30 September 2026

Read first: [Computer architecture](#), [Data storage](#)

© 2026 Pastopic. Free for personal study. Please share the link, not the file.

What you need to know

By the end of this topic you should be able to:

1. **Describe the difference between system software and application software**, and give examples of each. System software provides the services the computer needs (the operating system and utility software); application software provides the services the user needs.
2. **Describe what an operating system is for and what it does**: managing files, handling interrupts, providing an interface, managing peripherals and drivers, managing memory, managing multitasking, providing a platform for running applications, providing system security and managing user accounts.
3. **Understand how hardware, firmware and the operating system work together to run applications**: applications run on the operating system, the operating system runs on the firmware, and the bootloader (firmware) runs on the hardware.
4. **Describe what interrupts are for and how they work**: how an interrupt is generated, how it is handled by an interrupt service routine, and what happens as a result. Know examples of software interrupts (such as division by zero, or two processes trying to access the same memory location) and hardware interrupts (such as pressing a key on the keyboard, or moving the mouse).

The fetch–decode–execute cycle and the registers are in computer architecture; ROM, RAM and virtual memory are in data storage. Both are recapped here where an interrupt or memory question needs them.

Understand it

What software is

Software is the programs, the sets of instructions, that make a computer's hardware do something.

Hardware is the physical parts: the CPU, memory, storage and the devices plugged in. Software comes in two broad kinds.

System software and application software

System software provides the services the computer requires. It runs and looks after the computer itself, and gives other programs somewhere to run. There are two kinds:

- the **operating system (OS)**, such as Windows, macOS, Linux, Android or iOS;
- **utility software**, small programs that maintain or protect the computer: anti-virus, a software firewall, defragmentation, backup, compression and disk formatting software.

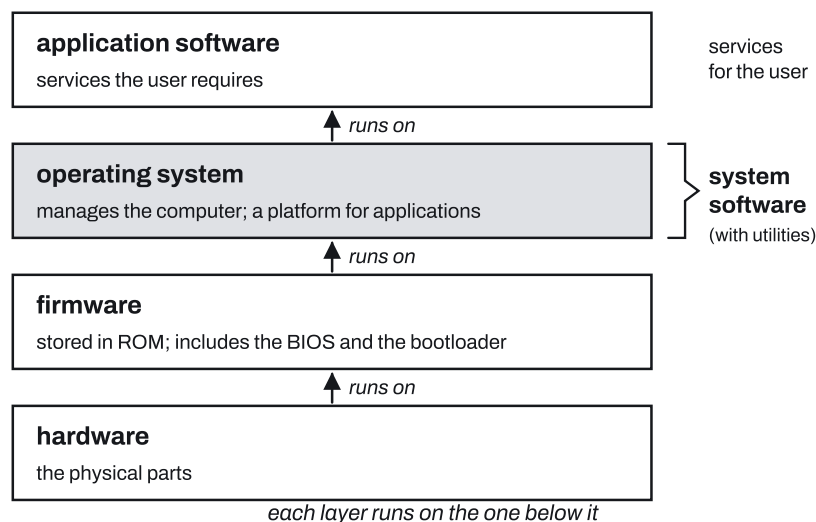
Application software provides the services the user requires. It lets the user carry out a task: a word processor, spreadsheet, database, web browser, photo editor, video editor, email client, music player, presentation software or a game.

A quick test: if the program is there so that **you** can get a job done (write an essay, edit a photo), it is application software. If it is there so that **the computer** keeps working (starting up, managing memory, removing viruses), it is system software.

In one line: system software provides the services the computer requires, whereas application software provides the services the user requires.

What runs on what

An application cannot talk to the hardware on its own. It needs layers underneath it.



- **Firmware** is software that is stored permanently in **read-only memory (ROM)**, so it stays there when the power is off. It controls the hardware at the lowest level. It includes the **BIOS** (basic input output system) and the **bootloader** (also called the **bootstrap**).
- When you switch the computer on, the hardware runs the bootloader. The bootloader checks the hardware and loads the operating system from secondary storage into RAM, then hands over to it.
- The operating system then manages everything, and gives applications a platform to run on.

So, from the bottom up: the bootloader (firmware) runs on the hardware; the operating system runs on the firmware; applications run on the operating system. (In a small embedded system, such as a washing machine, the firmware may be the only software there is.)

When the power is off, the operating system and applications are kept in **secondary storage** (a hard disk or SSD); only the firmware lives in ROM.

The operating system and its functions

The **operating system** is the system software that manages the whole computer: its hardware, its memory, its files and the programs that run on it. It sits between the user, the applications and the hardware. Its functions are:

Function	What it does
Managing files	Lets the user and programs create, open, save, copy, move, rename, delete and sort files, and organise them into folders
Handling interrupts	Gives each interrupt a priority, keeps the interrupt queue and calls the right interrupt service routine (below)
Providing an interface	Gives the user a way to control the computer, such as a graphical user interface (windows, icons, menus) or a command-line interface
Managing peripherals and drivers	Controls the input, output and storage devices; installs and uses each device's driver (the program that lets the OS talk to that device); sends data to and receives data from the hardware, using buffers to hold data on the way
Managing memory	Allocates memory to each process and frees it when the process ends; keeps track of which memory locations are in use; checks each process has enough memory; stops two processes using the same memory location; moves data between RAM and storage, including managing virtual memory
Managing multitasking	Lets several programs appear to run at once by giving each one a share of processor time and switching between them very quickly, using interrupts
Providing a platform for running applications	Applications run on the OS and use it to reach the hardware, so an application doesn't need to know what hardware it is on
Providing system security	Asks for a username and password (or biometrics) before anyone can use the computer, controls who can open which files, and keeps the system updated against threats
Managing user accounts	Lets several people have their own account on one computer, each with its own files, settings and access rights

You need to be able to **name** these and **describe** any of them. The written names above are the exact phrases the syllabus uses, so they are always safe answers. "Providing a platform for running applications" also shows up as "allows application software to run on the computer".

Interrupts

An **interrupt** is a signal sent to the CPU by a device or by a program to say that something needs the attention of the CPU. It lets the computer respond straight away to events, instead of waiting until it finishes what it is doing.

Hardware interrupts come from hardware devices:

- pressing a key on the keyboard;
- moving the mouse, or clicking a mouse button;
- plugging in a device, such as a USB stick, or unplugging one;
- a printer with a paper jam, or a printer that is out of paper or out of ink;
- a printer's buffer being empty, so it needs more data;
- the battery getting low, a component overheating, or a hardware fault.

Software interrupts come from a program:

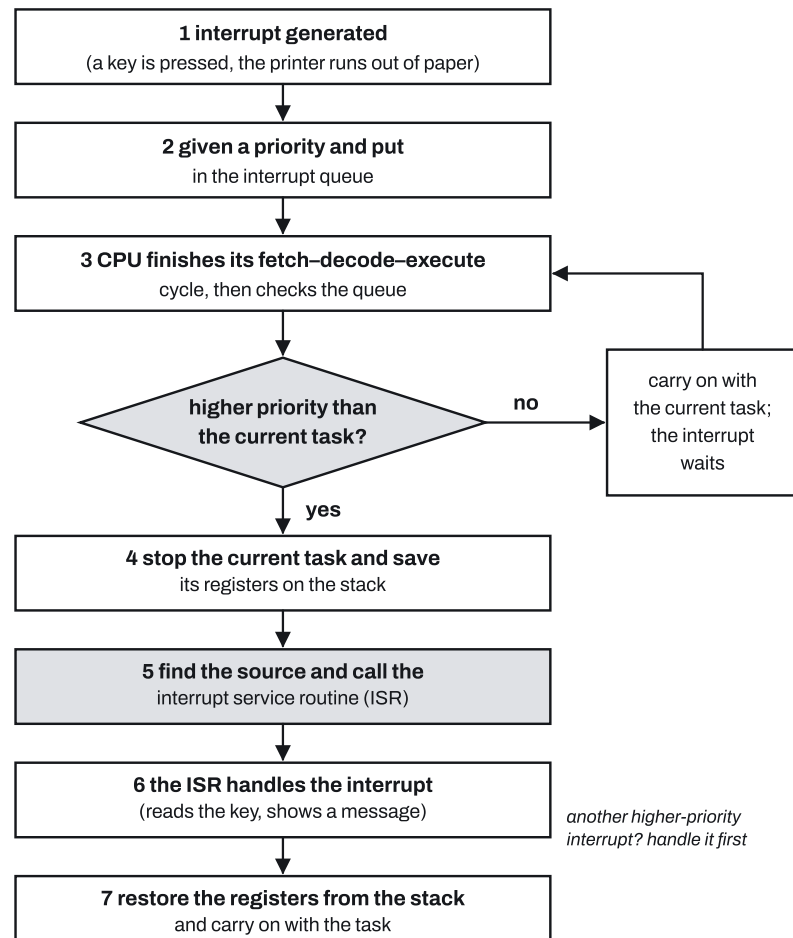
- division by zero;
- two processes trying to access the same memory location;
- a program that crashes or has an error (for example it tries to use a value that doesn't exist);
- switching from one application to another, or opening a new one.

Name the **event**, not just the device: "the keyboard" is a device, "a key is pressed on the keyboard" is an interrupt.

Why interrupts are needed. Without them the CPU would only start a new task when it had finished the current one. It could not multitask, errors would not be dealt with, and it would not react when you press a key or the printer runs out of paper: the computer would be almost impossible to use. Interrupts let the computer respond to urgent events, deal with them in order of importance, and switch between tasks.

How an interrupt is handled

The operating system and CPU follow the same steps every time.



1. **Generated.** A device or program sends an interrupt signal to the CPU (a key is pressed; the printer runs out of paper).
2. **Queued.** The interrupt is given a **priority** and placed in the **interrupt queue**, in order of priority. An urgent event, such as a hardware failure, has a high priority.
3. **Checked.** The CPU finishes the fetch–decode–execute cycle it is on. At the end of each cycle it checks the queue for an interrupt with a higher priority than the task it is running.
4. **If the interrupt has a lower priority**, the CPU carries on with the current task, and the interrupt waits in the queue.
5. **If it has a higher priority**, the CPU stops the current task and saves where it was (the contents of its registers, such as the program counter) on a **stack**, so that it can go back to it later.
6. The CPU checks the **source of the interrupt** (which device or program sent it) and calls the matching **interrupt service routine (ISR)**, also called the **interrupt handler**: a short program that deals with that kind of interrupt.
7. **The ISR handles the interrupt.** This is the result the user sees: the key's character appears on screen, a "paper jam" message is shown, more data is sent to the printer's buffer, or a program that divided by zero is stopped with an error.
8. **Carry on.** If another interrupt with a higher priority is waiting, it is handled next. Otherwise the CPU restores (retrieves) the saved registers from the stack and continues the task exactly where it stopped.

The interrupt itself is **not** a message to the user. It is a signal to the CPU. A message on the screen is only one possible **result** of handling it.

Key facts and formulas

There are no formulas in this topic. These are the facts to learn; nothing here is given in the exam.

Fact	Status
System software provides the services the computer requires: the operating system and utility software	Learn it
Application software provides the services the user requires, to perform tasks	Learn it
Utility examples: anti-virus, firewall, defragmentation, backup, compression, disk formatting	Learn it
Firmware: stored in ROM; includes the BIOS and the bootloader (bootstrap)	Learn it
From the bottom up: hardware → firmware (bootloader) → operating system → applications, each running on the one before	Learn it
The 9 OS functions: files, interrupts, interface, peripherals and drivers, memory, multitasking, platform for applications, security, user accounts	Learn it
Interrupt: a signal to the CPU that something needs its attention	Learn it
Hardware interrupts (key press, mouse move or click, printer out of paper) and software interrupts (division by zero, two processes using one memory location)	Learn it
Handling: priority → queue → check after each cycle → save registers on stack → ISR → restore	Learn it

How to do it

In the Paper 1 questions we have tagged for this topic (22 questions, 2021–2025), the types came up this often. 13 of the 22 questions mix more than one type, so the counts add up to more than 22.

Question type	Questions
Interrupts: what they are, why they are needed and examples	11
Functions of the operating system	9
How an interrupt is handled	7
System software and application software	6

Question type	Questions
Firmware, the bootloader and the layers of software	5

1. Interrupts: what they are, why they are needed and examples (11 questions)

1. **Name the signal.** When a question describes a signal that stops the music when a call comes in, that tells the computer the printer buffer is ready for more data, or that lets multitasking happen, the answer is **interrupt**.
2. **Name the type.** An interrupt from a device (a key press, a mouse click) is a **hardware interrupt**; one from a program (division by zero) is a **software interrupt**.
3. **Give examples.** Write each as an **event**: "a key is pressed", "the mouse is moved", "the printer runs out of ink", "a USB stick is plugged in", "a program tries to divide by zero".
4. **State its purpose** (1 mark): to tell the CPU that something needs the attention of the CPU.

Variations you will meet:

- **"Give two other examples"** after the question has used one (a paper jam, a key press). Don't repeat it, and don't give two that are really the same. Examiners say most candidates only give printer examples: use a range, such as a key press, a mouse click and division by zero.
- **Examples in a setting**, such as a factory robot: user input (someone presses a button), an emergency stop, the robot meets an obstacle, the battery is low, a component overheats, a hardware or software error.
- **"State what would happen if interrupts were not used"**: the computer would only start a new task when it had finished the current one, so it could not multitask, and errors would not be dealt with.
- **"Identify one input device and an interrupt it generates"**: the pair must match: keyboard and a key is pressed; mouse and the mouse is moved or a button is clicked.
- **"Give the name of the type of software that manages interrupts"**: the operating system (or, for the program itself, the interrupt handler).

2. Functions of the operating system (9 questions)

1. **Name the software.** "The software that manages files, memory and multitasking" or "that manages inputs and outputs" is the **operating system** (system software).
2. **Give functions** (1 mark each): choose from the nine in the table. Don't repeat the one the question gives you.
3. **Describe a function** (usually 1 mark to name it, 1 to describe it): say what the OS actually **does**, from the table. "Managing memory: it allocates memory to each process."

Variations you will meet:

- **"Describe the role of the OS in managing memory"** (2–3 marks). Examiners say this is poorly answered, so learn the points: it allocates memory to processes and frees it afterwards; it checks each process has enough memory; it stops two processes using the same memory location; it keeps track of

which locations are in use; it moves data between RAM and storage and manages virtual memory, so that memory is used efficiently.

- **"Give two functions that relate to memory"**: managing memory, and managing multitasking (several processes share memory).
- **"Give one task the OS lets the user do to manage files"**: create, open, save, copy, move, rename, delete or sort a file.
- **A tick box of functions, one of which is not**: "loading the bootstrap" is done by firmware, not the OS; a job for an application (editing a photo, compressing images for a website) is also not an OS function.
- **A table of function names and descriptions**: match them. "Allows application software to run on the computer" is providing a platform for running applications; "transmits data to and receives data from hardware" is managing peripherals.

3. How an interrupt is handled (7 questions)

This is the longest answer in the topic (4–7 marks). Write the steps in order, one per mark:

1. the device or program **generates** the interrupt and sends it to the CPU;
2. the interrupt is given a **priority** and placed in the **interrupt queue**;
3. the CPU finishes its current **fetch–decode–execute cycle** and checks the queue;
4. if the interrupt's priority is **higher** than the current task, the CPU **stops** the current task;
5. it **saves** the current task's registers on the **stack**;
6. it checks the **source** of the interrupt and calls the **interrupt service routine (ISR)** (interrupt handler);
7. the ISR **handles** the interrupt, for example it shows the message;
8. then the CPU **restores** the registers from the stack and carries on with the task (unless another higher-priority interrupt is waiting);
9. if the priority is **lower**, the CPU carries on with the current task and the interrupt waits.

Variations you will meet:

- **"Describe the role of the OS in handling interrupts"** (2 marks): it gives each interrupt a priority; it maintains the interrupt queue; it uses the interrupt service routine to process the interrupt. Examiners say few candidates got past "priority": learn all three.
- **A setting to explain**, such as a key press on a keyboard or a printer that jams and shows a message. Use the same steps, but name the device and the result: "the printer generates an interrupt ... the ISR outputs a message that the paper has jammed".
- **Starting part-way through**, when the question says the interrupt is already queued (as in [0478/12/O/N/25 Q5](#)): start from step 3 (check the priority) and include both branches: higher → stop, stack, source, ISR; lower → carry on.
- **A paragraph with gaps and a word list**: the words are usually printer or device, computer or CPU, priority level, fetch–decode–execute cycle, interrupt queue, higher, interrupt service routine (ISR).
- **Name the program that deals with an interrupt**: interrupt service routine (ISR), or interrupt handler.

4. System software and application software (6 questions)

1. **Describe the difference** (2 marks): system software provides the services the computer requires (it runs and manages the hardware and software), **whereas** application software provides the services the user requires (it lets the user perform tasks).
2. **Examples** (1 mark each). System software: operating system, or a named utility such as anti-virus or defragmentation software. Application software: word processor, spreadsheet, database, web browser, photo editor, video editor, email client, game.

Variations you will meet:

- **"Describe the difference, giving an example of each"** (4 marks): two marks for the difference (both sides), one for a system example, one for an application example.
- **"Tick the example of utility software"**: anti-virus (or defragmentation, backup, compression). Spreadsheets, web browsers and databases are application software.
- **"State what is meant by software"**: the programs or instructions that make the computer's hardware operate.
- **"Which component stores both types when the power is off?"**: secondary storage (a hard disk drive or SSD), not RAM.
- **"Give two examples of system software"**: operating system and utility software, or two named utilities.

5. Firmware, the bootloader and the layers of software (5 questions)

1. **What firmware is** (1 mark): software (instructions) permanently stored in ROM; it controls the hardware and provides the operating system with a platform to run on.
2. **Examples of firmware**: the bootloader (bootstrap) and the BIOS; in an embedded system, its whole program.
3. **The layers**: applications run on the operating system, the operating system runs on the firmware, and the bootloader (firmware) runs on the hardware.

Variations you will meet:

- **Name the software the OS needs underneath it to start**: firmware (or bootloader, bootstrap or BIOS).
- **A terms table**: match each description to hardware, application software, operating system or firmware. Look for the clue: physical parts mean hardware; doing a job for the user means application software; running the computer (files, memory) means the operating system; kept in ROM means firmware.
- **Statements to complete from a list**, about the layers from the bottom up. Examiners say "bootloader" was often written in the wrong gap: the bootloader **is** firmware, so the layer under firmware must be hardware.

Worked examples

The bold codes show where marks are earned, as in Cambridge mark schemes. The number is how many marks.

- **B1** mark for a correct result on its own, with no method needed.
-

Example 1

A student's laptop has these programs installed: an operating system, a music player, anti-virus software and a spreadsheet.

(a) Tick (✓) **one** box to show which of these is an example of utility software. **[1]**

	Software
A	photo editor
B	defragmentation software
C	web browser
D	game

(b) Identify the **two** programs on the student's laptop that are system software. **[2]**

(c) Describe the difference between system software and application software. **[2]**

(a) B **B1**

(b) The operating system **B1** and the anti-virus software **B1**.

(c) System software provides the services that the computer requires, such as managing the hardware **B1**, whereas application software provides the services that the user requires, letting them carry out tasks such as playing music **B1**.

Example 2

A school's computers each have an operating system.

(a) One function of the operating system is managing user accounts. Give **two other** functions of an operating system. **[2]**

(b) Describe the role of the operating system in managing memory. **[3]**

(c) Identify **one** function of the operating system that lets a student use a new printer, and describe what it does. **[2]**

(a) Any two: managing files **B1**; providing an interface **B1**; handling interrupts; managing multitasking; providing system security; providing a platform for running applications.

(b) Any three:

- it allocates memory to each process, and frees it when the process ends **B1**;
- it makes sure two processes don't try to use the same memory location **B1**;
- it checks that each process has enough memory **B1**;
- it moves data between RAM and secondary storage (virtual memory) so memory is used efficiently.

(c) Managing peripherals and drivers **B1**. It installs and uses the printer's driver so the computer can communicate with the printer, and sends the print data to it (through a buffer) **B1**.

Example 3

Aroha is editing a video when she plugs a USB memory stick into her computer. An interrupt is generated.

(a) State what is meant by an interrupt. **[1]**

(b) State whether this is a hardware interrupt or a software interrupt. **[1]**

(c) Give **one** example of the other type of interrupt. **[1]**

(d) Explain how the interrupt is handled. **[5]**

(a) A signal sent to the CPU to show that something needs the attention of the CPU. **B1**

(b) Hardware interrupt. **B1**

(c) Division by zero (or two processes trying to access the same memory location). **B1**

(d) Any five:

- the interrupt is given a priority **B1**;
- and placed in the interrupt queue **B1**;
- the CPU finishes its current fetch–decode–execute cycle and checks the queue **B1**;
- the interrupt has a higher priority than the video editing, so the CPU stops the current task **B1**;
- and saves its registers on the stack **B1**;
- it checks the source of the interrupt (the USB port) and calls the interrupt service routine;
- the ISR handles the interrupt, so the memory stick appears in the file manager;
- the CPU restores the registers from the stack and carries on with the video editing.

(If the interrupt had a lower priority, the CPU would carry on editing and leave the interrupt in the queue.)

Example 4

Complete the statements about how a computer starts up and runs software. Use the terms from the list. Not all the terms will be used. Use each term once only. [5]

application · bootloader · compiler · hardware · operating system · random access memory (RAM) · read-only memory (ROM) · utility

The physical components of the computer are called the Firmware is stored in so that it is not lost when the power is off. When the computer is switched on, the firmware called the is run. It loads the , which is then run on the firmware. A word processor is software, which runs on the operating system.

Answers, one mark each: hardware **B1**; read-only memory (ROM) **B1**; bootloader **B1**; operating system **B1**; application **B1**.

Common mistakes

- **Mixing up the layers.** The bootloader is firmware, so "the operating system is run on the firmware, which is run on the **hardware**", not on the bootloader. Examiners saw "bootloader" in the wrong gap often.
- **Not knowing what firmware is.** Examiners say few candidates could define it. It is software permanently stored in ROM that controls the hardware and starts up the computer (the BIOS and the bootloader).
- **Giving a device, not an interrupt.** "Keyboard" or "mouse" earns nothing as an example; "a key is pressed on the keyboard" does.
- **Only printer examples.** Nearly every example candidates give is about printing. Use a range: key press, mouse click, USB device plugged in, division by zero.
- **Thinking an interrupt is a message to the user.** The interrupt is a signal to the CPU. An error message on the screen is only a result of the ISR handling some interrupts.
- **Giving an example when asked why interrupts are needed.** "Why" wants the general reason: so the CPU can stop what it is doing to deal with something urgent, which allows multitasking and lets errors be dealt with.
- **Stopping at "priority".** When describing how interrupts are handled, most candidates only say the interrupt gets a priority. Add the queue, the check after each cycle, stopping the task, the stack, the ISR, and restoring the task.
- **Vague memory management.** "It manages the memory" repeats the question. Say what it does: allocates memory to processes, stops two processes using the same location, moves data between RAM and storage.
- **Calling an application "utility software".** A web browser or spreadsheet is application software; utility software maintains the computer (anti-virus, defragmentation, backup).
- **Repeating the function the question gave you**, such as "handling interrupts" when it says "one function is handling interrupts; give two others".

How to get the marks

- **"Identify", "give", "state"** (1 mark each): the exact term: "operating system", "hardware interrupt", "interrupt service routine", "firmware", "bootloader". A short, precise term beats a sentence.
- **"Describe the difference"** (2 marks): both sides, joined by "whereas": what the computer needs **whereas** what the user needs. One side alone earns one mark at most.
- **"Describe" or "explain" how an interrupt is handled** (4–5 marks): one step per mark, in order, using the key words: priority, queue, fetch–decode–execute cycle, stop, stack, source, ISR, restore. If the question gives a device, name it and its result.
- **Function + description**: the description must say what the OS **does**, not reword the name. "Managing memory: manages memory" earns only the name mark.
- **Examples**: each must be different, and written as an event. Read the question: if it has used paper jam, pick something else.
- **Tick boxes**: tick exactly one box. "Which is not a function of the OS" wants the one done by something else (firmware or an application).
- **Word-list gap fills**: use each word once; check each finished sentence makes sense on its own.
- **Read the setting**. A robot, a printer or a phone call tells you which examples will fit: give interrupts that could happen there.

Quick check

1. Tick (✓) **one** box to show which is **not** a function of an operating system. A managing memory; B providing an interface; C compressing images for a website; D managing user accounts.
2. State whether each is a hardware or a software interrupt: (a) a scanner runs out of paper; (b) a program divides by zero; (c) the mouse is moved; (d) two programs try to write to the same memory location.
3. State what is meant by firmware, and give **one** example.
4. A processor is running a program when it receives an interrupt with a lower priority than the program. State what the processor does.
5. Give **two** things that could not happen if a computer did not use interrupts.

Answers

1. C. Compressing images for a website is a task the user does with application (or utility) software; the other three are OS functions.
2. (a) hardware; (b) software; (c) hardware; (d) software.
3. Software (instructions) stored permanently in ROM that controls the hardware and starts up the computer, providing the operating system with a platform to run on. Example: the bootloader (bootstrap) or the BIOS.
4. It carries on with the program (running its next fetch–decode–execute cycle), and the interrupt waits in the interrupt queue until nothing with a higher priority is running.
5. Any two: the computer could not multitask; it would only start a new task after finishing the current one; errors would not be dealt with; it would not respond to key presses or devices straight away.

Past questions to try

- **0478/13/M/J/24 Q9**: complete the paragraph on how a printer's interrupt is handled, then two memory-related OS functions.
- **0478/11/O/N/23 Q9**: how interrupts are used when a key is pressed, then hardware and software examples.
- **0478/11/O/N/23 Q3**: a terms table of hardware, application software, operating system and firmware.
- **0478/13/M/J/23 Q11**: the operating system's file and memory management, and the signal that allows multitasking.