

1. Which is a core job of an operating system?
 - ☐ writing the user's documents
 - ☐ manufacturing the hardware
 - ☐ designing the CPU chip
 - ☐ managing the hardware so programs can share it safely
2. A program library is:
 - ☐ a folder of the user's documents
 - ☐ the operating system kernel
 - ☐ pre-written code (subroutines, classes) that programs reuse
 - ☐ a kind of CPU register
3. A compiler differs from an interpreter because it:
 - ☐ translates the whole program once and produces a stand-alone executable
 - ☐ never reports errors
 - ☐ runs the program one line at a time
 - ☐ must be installed to run the finished program
4. The operating system lets several programs share the same hardware safely, so they don't interfere with each other.
 - ☐ True ☐ False
5. A collection of ready-made, reusable, tested code is a program _____.

6. When does an interpreter report an error?
 - ☐ never
 - ☐ before the program starts
 - ☐ when it reaches the line containing the error
 - ☐ only after the whole program is translated
7. Which are purposes of an operating system? Select **all** that apply.
 - ☐ sharing processor time, memory and devices between programs
 - ☐ providing an interface between the user and the hardware

- ☐ translating high-level source code into machine code
- ☐ hiding the complexity of the hardware from application programs

Tick every correct option.

8. A benefit of using a library is that the code is:

- ☐ well-tested and reliable, saving development time
- ☐ specific to one program only
- ☐ always smaller than writing it yourself
- ☐ impossible to update

9. An interpreter produces no stand-alone executable, so the interpreter itself must be installed to run the program.

- ☐ True ☐ False

10. System-maintenance programs such as antivirus, backup and defragmenter are called _____ software.

1. managing the hardware so programs can share it safely

The OS manages hardware and provides services and a UI, letting multiple programs run safely on shared hardware.

2. pre-written code (subroutines, classes) that programs reuse

Libraries provide ready-made, tested code (maths, graphics, network...) that a program can call instead of writing its own.

3. translates the whole program once and produces a stand-alone executable

A compiler translates all at once into an executable that runs without the compiler. An interpreter runs line by line.

4. True

Sharing the CPU, memory and devices safely between many programs is exactly what the OS is for.

5. library

Libraries save time and reduce bugs.

6. when it reaches the line containing the error

An interpreter runs line by line, so it reports an error at the moment it reaches that line – handy during development.

7. sharing processor time, memory and devices between programs; providing an interface between the user and the hardware; hiding the complexity of the hardware from application programs

Interface, hiding complexity, managing and sharing resources are the OS's jobs. Translation is a compiler's or interpreter's.

8. well-tested and reliable, saving development time

Library code is tested, widely used and standardised – reusing it saves time and improves reliability.

9. True

An interpreter translates and runs line by line each time – there is no separate executable to ship, unlike a compiler.

10. utility

Utility software maintains the system; word processors and browsers are application software.