

Chapter 5

Software System

Operating Systems • Utility Programs • Device Drivers • Application Software

Free Notes for Class 9 Computer Science Students



Chapter 5: Software System

This chapter explores software — the collection of programs and instructions that tell a computer what to do — as the essential intermediary between users and computer hardware. It distinguishes system software, which manages hardware and provides a platform for other programs to run, from application software, which helps users perform specific tasks.

The chapter examines the operating system in detail, covering how it manages hardware resources, provides user interfaces (GUI and CLI), and runs applications, along with utility programs (disk cleanup, antivirus, backup, file compression) and device drivers. It closes by surveying common categories of application software — word processing, spreadsheets, and graphic design — with real-world examples of each.

Learning Objectives

- Identify and explain the significance of system software and application software
- Understand the role and main functions of system software, including operating systems
- Explain how operating systems manage hardware resources, provide user interfaces, and run applications
- Describe how utility software enhances system performance, security, and maintenance
- Understand how device drivers facilitate communication between hardware devices and the operating system
- Recognize the main functions of commonly used application software (word processing, spreadsheets, graphic design)
- Differentiate between system software and application software in terms of roles and functions
- Identify appropriate software tools for specific tasks based on their functions and capabilities



Key Concepts

5.1 Software and Its Two Main Types

Software is a collection of programs and instructions that tell a computer what to do and how to do it; without software, computers would be little more than useless machines. Software is broadly divided into two categories: system software and application software.

System software is designed to manage system resources and provide a platform for application software to run, acting as a bridge between hardware and user applications — examples include operating systems (Windows, macOS, Linux), device drivers (printer, graphics card, sound card drivers), and utility programs (antivirus software, disk cleanup tools, backup software). Application software is designed to help users perform specific tasks and is typically far more varied than system software — examples include word processors (Microsoft Word, Google Docs), web browsers (Chrome, Firefox), games, and media players.

5.2 The Operating System: An Overview

An Operating System (OS) is a type of system software that manages all the hardware and software on a computer, acting as an intermediary between the computer hardware and user applications. It ensures that different programs and users do not interfere with each other, and provides a stable, consistent way for applications to interact with hardware without needing to know all its technical details.

The most commonly used operating systems are: Windows (Microsoft's popular OS for personal computers, with a start menu, taskbar, and application windows), macOS (Apple's OS for Mac computers, with a dock and features like Mission Control), Linux (an open-source OS used from servers to desktops, appearance varying by distribution), Android (Google's OS for smartphones and tablets), and iOS (Apple's OS for iPhones and iPads, known for smooth performance).

5.3 Operating System Function: Managing Hardware Resources

One of the primary functions of an operating system is managing the hardware resources of a computer system — including the CPU, memory, disk drives, and



peripheral devices like printers and keyboards. The OS ensures each application receives the necessary resources to function correctly without interfering with other applications.

For example, when you open a web browser while listening to music, the operating system allocates CPU time and memory to both the web browser and the music player, ensuring both run smoothly by managing resources effectively — a task that becomes increasingly complex as more applications run simultaneously.

5.4 Operating System Functions: User Interfaces and Running Applications

The operating system provides a User Interface (UI) that allows users to interact with the computer, in two main forms. A Graphical User Interface (GUI) allows users to interact using visual elements like windows, icons, and menus — user-friendly and intuitive (used by Windows and macOS: click icons, drag-and-drop files, use menus). A Command-Line Interface (CLI) requires users to type text commands to perform tasks — more flexible and powerful, but harder for beginners (used by Linux and DOS: typing commands to copy files, run programs, configure settings).

The operating system is also responsible for running applications: it loads applications into memory, allocates necessary resources, and manages their execution, ensuring applications don't interfere with each other and run efficiently. For example, when you open a word processor, the OS loads it into memory and allocates CPU time; if multiple applications are open, the OS manages resource distribution so all can run simultaneously without performance issues.

5.5 Utility Programs

Utility programs are essential components of system software that enhance a computer's functionality by performing tasks that ensure smooth operation and efficient management of hardware, software, and data. Disk Cleanup scans the hard drive for temporary and cached files that can be safely deleted, reclaiming disk space and potentially improving performance. Antivirus Software scans files



and incoming data for known viruses and malware, providing real-time protection against attacks.

Backup Software schedules regular backups of files and folders to external drives, cloud storage, or network locations, allowing full system or selective file backups — useful for recovering an accidentally deleted file. File Compression Tools compress one or more files into a single archive format (like ZIP or RAR) while preserving data integrity, reducing file size to make storage and transfer faster and easier, and often providing encryption and password protection.

5.6 Device Drivers

Device drivers facilitate communication between hardware devices and the operating system, ensuring devices function correctly — acting like a translator between the computer and its gadgets (printer, keyboard, mouse, graphics card, and so on). A printer driver helps the computer send the correct signals to the printer so it can print documents; a graphics card driver ensures the computer can display images and videos correctly on the screen.

Device drivers work in three stages: Installation (when a new device is connected, its driver is often installed), Communication (the driver translates general instructions from the computer into specific instructions the device understands), and Operation (once installed, the driver helps the computer and device work together smoothly). A useful analogy is a TV remote control: the TV (device) can change channels and adjust volume but needs instructions; the remote control (driver) sends the correct signals to the TV; you (the computer) decide what to do and use the remote to communicate it. A Plug and Play (PnP) device automatically configures itself when connected, simplifying installation and use.

5.7 Application Software: Word Processing and Spreadsheets

Word processing software is used for creating, editing, formatting, and printing documents — essential for writing letters, reports, and essays. Examples include Microsoft Word (widely used, with text formatting, spell/grammar check, and image/table/chart insertion), Google Docs (web-based, real-time collaboration), Apple Pages (macOS/iOS, templates and design tools), and LibreOffice Writer (free, open-source, cross-platform).



Spreadsheet software organizes, analyzes, and stores data in tabular form, using a grid of cells arranged in rows and columns for calculations, data input, and charts — essential for budgeting, financial analysis, and statistics. Examples include Microsoft Excel (complex formulas, pivot tables, charts), Google Sheets (web-based, real-time collaboration), Apple Numbers (macOS/iOS, strong visualization tools), and LibreOffice Calc (free, open-source alternative to Excel).

5.8 Application Software: Graphic Design

Graphic design software is used for creating, editing, and managing visual content — providing tools for drawing, painting, photo editing, and illustration, essential for designers and artists across advertising, web design, publishing, and multimedia. Examples include Adobe Photoshop (photo editing and digital painting), Adobe Illustrator (vector graphics for logos and scalable illustrations), CorelDRAW (user-friendly vector graphics editor), GIMP (free, open-source alternative to Photoshop), and Canva (web-based, template-driven, beginner-friendly design tool).

AI-based tools are increasingly integrated across these application software categories: Grammarly and Microsoft Editor enhance word processing with grammar and style suggestions; Microsoft's Ideas in Excel and Google Sheets' Explore feature assist spreadsheet analysis; and Adobe Sensei automates repetitive tasks and enhances images in graphic design software.

Important Definitions

What is software?

A collection of programs and instructions that tell a computer what to do and how to do it; without software, computer hardware would be useless.

What is system software?

Software designed to manage a computer's system resources and provide a platform for application software to run, acting as a bridge between hardware and user applications.

What is application software?

Software designed to help users perform specific tasks, such as word processing, web browsing, or gaming, built to fulfil particular user needs.



What is an Operating System (OS)?

A type of system software that manages all the hardware and software on a computer, acting as an intermediary between computer hardware and user applications.

What is a Graphical User Interface (GUI)?

A user interface that allows users to interact with a computer using visual elements such as windows, icons, and menus, making it user-friendly and intuitive.

What is a Command-Line Interface (CLI)?

A user interface that requires users to type text commands to perform specific tasks; more flexible and powerful than a GUI, but harder for beginners to use.

What is a utility program?

A component of system software that performs tasks to ensure smooth operation and efficient management of hardware, software, and data, such as disk cleanup or antivirus software.

What is a device driver?

A piece of system software that facilitates communication between a hardware device and the operating system, translating general instructions into device-specific commands.

Key Facts and Relations

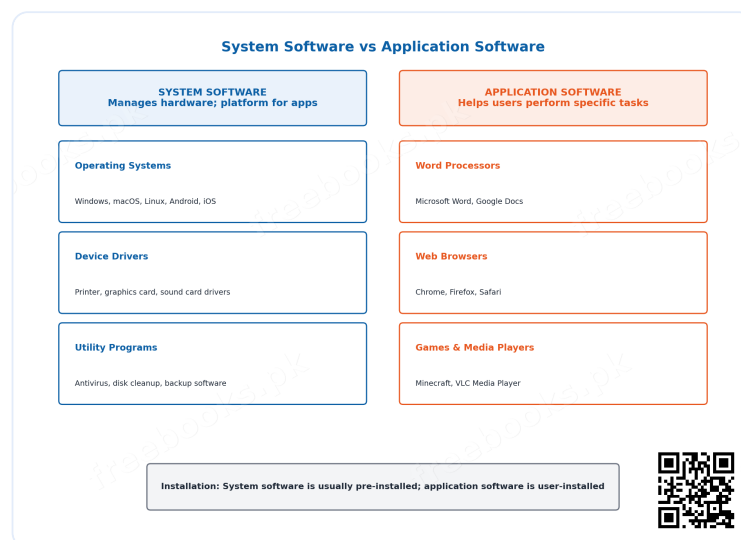
Topic	Key Fact / Relation
Software's two main types	System Software (manages hardware) + Application Software (performs user tasks)
Examples of system software	Operating Systems, Device Drivers, Utility Programs
Examples of application software	Word Processors, Web Browsers, Games, Media Players
Two types of user interface	GUI (Graphical User Interface) and CLI (Command-Line Interface)
3 main OS functions	



	Managing Hardware Resources + Providing a User Interface + Running Applications
4 common utility programs	Disk Cleanup, Antivirus Software, Backup Software, File Compression Tools
How device drivers work (3 stages)	Installation → Communication → Operation
5 major operating systems	Windows, macOS, Linux, Android, iOS

Diagrams

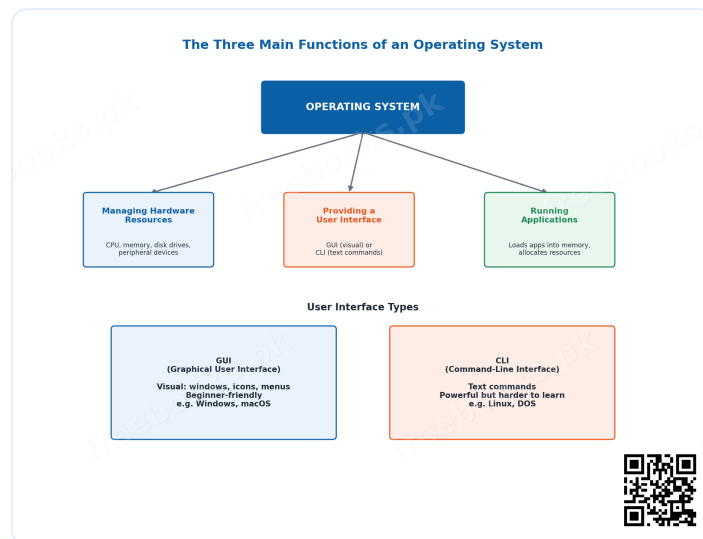
System Software vs Application Software: A comparison of system software (operating systems, device drivers, utility programs) and application software (word processors, browsers, games) with their purposes and examples



System Software vs Application Software

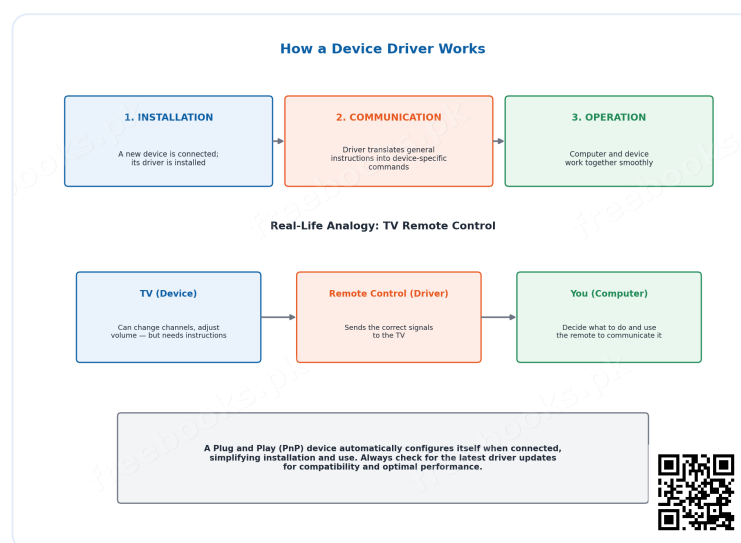
The Three Main Functions of an Operating System: A diagram showing how an operating system manages hardware resources, provides a user interface (GUI vs CLI), and runs applications





The Three Main Functions of an Operating System

How a Device Driver Works: A diagram of the three-stage device driver process (Installation, Communication, Operation) using the TV remote control analogy



How a Device Driver Works

Short Questions & Answers

Define system software and provide two examples.

System software manages a computer's hardware resources and provides a platform for application software to run. Examples include operating systems (like Windows) and device drivers (like a printer driver).

Explain the primary functions of an operating system.



An operating system's primary functions are managing hardware resources (CPU, memory, devices), providing a user interface (GUI or CLI) for user interaction, and running applications by loading them into memory and allocating resources.

What is utility software and why is it important?

Utility software performs tasks that maintain and enhance a computer's performance, security, and reliability, such as disk cleanup, antivirus scanning, backups, and file compression; it is important because it keeps a system running efficiently and protects it from threats.

Describe the role of device drivers in a computer system.

Device drivers act as translators between hardware devices and the operating system, converting general instructions from the computer into specific instructions the device can understand, allowing devices like printers and graphics cards to function correctly.

Differentiate between system software and application software with examples.

System software (e.g., Windows, device drivers) manages hardware and provides a platform for other software to run; application software (e.g., Microsoft Word, Google Chrome) helps users perform specific tasks like writing or browsing the web.

What are the main functions of spreadsheet software?

Spreadsheet software organizes, analyzes, and stores data in a grid of rows and columns, allowing users to perform calculations, create charts, and manage data for tasks like budgeting and statistical analysis.

How can graphic design software be used in the field of education?

In education, graphic design software can be used to create visual learning materials, infographics, presentation graphics, posters for school events, and illustrations that make lessons more engaging and easier to understand.

What is the difference between a GUI and a CLI?

A GUI (Graphical User Interface) lets users interact visually using windows, icons, and menus, making it beginner-friendly; a CLI (Command-Line Interface) requires typing text commands, offering more power and flexibility but a steeper learning curve.



Long Questions & Answers

Discuss the importance of system software in a computing system, explaining its main functions and why a computer cannot function without it.

System software occupies an absolutely foundational and truly indispensable position within any modern computing system, serving as the essential and continuously active intermediary layer that sits directly between a computer's raw physical hardware components and the various different application programs that a user actually wants to run and directly interact with on a day-to-day basis; without this essential underlying layer of system software running continuously in the background, a computer's hardware components — its processor, its memory chips, its storage drives, and its many other physical components — would simply remain nothing more than a collection of disconnected, disorganized electronic parts, entirely incapable of performing any genuinely useful, coordinated work whatsoever. The single most critical and central piece of system software found within virtually any computing device is its operating system, and this operating system performs three main, absolutely essential functions that collectively make the entire computer usable and productive for its human user. The first of these three essential functions is managing hardware resources: a modern computer's processor, memory, storage drives, and various peripheral devices must all be very carefully and continuously allocated among the multiple different application programs that frequently need to run simultaneously at any given moment, and it is specifically the operating system's job to fairly and efficiently distribute these limited physical resources so that, for example, a user is able to comfortably browse the web while simultaneously listening to background music, without either one of these two different running applications completely monopolizing the computer's available processing power or memory capacity. The second essential function is providing a user interface, whether that interface takes the form of a visually intuitive Graphical User Interface (GUI), employing icons, clickable menus, and draggable windows, or alternatively a more powerful but comparatively less beginner-friendly Command-Line Interface (CLI), requiring the user to directly type in specific text-based commands; without some form of properly functioning user interface being actively provided by the operating system, an ordinary human



user would have absolutely no practical, workable way to meaningfully communicate their specific wishes and intentions to the underlying computer hardware at all. The third essential function is running applications: the operating system is directly and continuously responsible for loading requested application programs into the computer's working memory, correctly allocating the necessary processing resources to each running application, and then carefully managing their execution moment-to-moment so that multiple separate applications are able to run together smoothly and reliably, all without any one of them causing serious interference with any of the others that happen to be running at the very same time. Beyond the central operating system itself, other important categories of system software, including various utility programs (such as antivirus software or disk cleanup tools) and individual device drivers (such as a specific printer driver), further significantly extend and enhance the underlying operating system's core underlying capabilities, collectively helping to maintain strong ongoing system security, reliable overall performance, and full functional compatibility with the very wide range of different hardware peripherals that users commonly wish to connect to their computers. Taken together as a complete, integrated whole, this entire foundational layer of essential system software is precisely what fundamentally transforms an otherwise inert, disconnected collection of physical electronic hardware components into a fully coherent, genuinely usable, and highly productive computing system, one capable of reliably running the enormously wide and diverse range of different application software that users everywhere have now come to depend upon so heavily and so consistently in their everyday personal and professional lives.

Describe the roles of operating systems, utility software, and device drivers, providing examples of each.

Although operating systems, utility software, and device drivers are all properly classified together as distinct component categories that collectively fall under the single broader umbrella term of 'system software,' each one of these three distinct categories actually plays a genuinely different, complementary role in keeping a modern computer system running smoothly, reliably, and securely on an ongoing basis, and understanding these various important differences helps to clarify precisely how a complete, fully functioning computer system is actually organized and structured internally. The operating system serves as the single



most central and comprehensive piece of system software running on virtually any given computer, acting as the primary, overarching intermediary directly between the computer's underlying physical hardware and the various different application programs that a user actively chooses to run; well-known, widely used examples of different operating systems include Microsoft Windows (a very popular, widely used choice specifically for personal computers), Apple's macOS (specifically designed and optimized for Apple's own line of Mac computers), Linux (a flexible, open-source operating system commonly used across an extremely broad range of devices, extending all the way from small home servers up to large-scale enterprise systems), Google's Android (specifically designed for smartphones and tablet devices), and Apple's iOS (specifically designed for iPhone and iPad devices); the operating system's own core responsibilities specifically include the ongoing management of hardware resources, the provision of an interactive user interface, and the reliable execution of application programs, as has already been discussed in detail elsewhere in this chapter. Utility software, by contrast, consists of an entire separate category of generally smaller, more narrowly specialized and highly focused programs that are each specifically designed to perform one single, particular maintenance-related or optimization-related task, working to further enhance a computer system's overall performance, security, or general reliability in some specific, well-defined respect; representative examples of different utility software programs include Disk Cleanup (a tool specifically used for identifying and then safely removing unnecessary temporary files in order to help free up valuable storage space), antivirus software (used for actively detecting and effectively removing any malicious software or malware that may have infected a system), backup software (used for creating and properly maintaining protective copies of a user's important files and data, guarding against potential future data loss), and dedicated file compression tools (used for reducing the overall physical size of one or more files, in order to make file storage and file transfer processes both faster and generally more convenient for the end user). Device drivers, finally, represent yet another distinctly different category of system software, one that is specifically and narrowly focused on enabling reliable, accurate communication between one single specific hardware device and the computer's operating system; a printer driver, for a clear example, is specifically and narrowly responsible for accurately translating a computer's generic print



commands into the very particular, highly specific signals that a specific individual printer model actually requires and correctly understands in order to properly and successfully print a given document, while a separate graphics card driver is instead specifically responsible for ensuring that visual images and video content are correctly rendered and displayed on a computer's connected screen or monitor. Considered altogether as a complete, integrated whole, these three genuinely distinct but very closely interconnected and cooperating categories of system software work together continuously, moment by moment, in order to reliably and consistently keep a modern computer system functioning smoothly, securely, and effectively on an ongoing, uninterrupted basis.

Explain the differences between system software and application software, covering purpose, examples, and installation.

System software and application software represent the two fundamental, clearly distinguishable categories into which essentially all computer software can be meaningfully classified and organized, and gaining a genuinely clear, thorough understanding of the various important differences that exist between these two distinct categories is absolutely essential for properly and fully understanding precisely how a complete, fully functioning computer system is actually organized and structured internally at a fundamental level. In terms of their underlying core purpose, system software is very specifically and narrowly designed to manage and directly operate a computer's underlying physical hardware components, and, in doing so, it thereby makes it genuinely possible for various different application software programs to subsequently run correctly and reliably on top of that same underlying hardware; put in slightly different but essentially equivalent terms, system software's core underlying purpose is fundamentally to keep the computer itself operating correctly, smoothly, and reliably at a basic technical level. Application software, by clear contrast, is instead very specifically designed and built with the different underlying purpose of directly helping the actual human user of the computer to successfully accomplish one or more particular, well-defined tasks that they specifically want or need to carry out — tasks such as writing a formal letter, carefully analyzing a detailed financial budget, or simply browsing a favourite website; application software's core underlying purpose, in other words, is fundamentally to help the human user themselves get some particular kind of genuinely useful, meaningful



work actually done using the computer. In terms of the kinds of concrete, specific examples that fall under each of these two categories, system software specifically includes examples such as operating systems (like Microsoft Windows or Apple's macOS) and individual device drivers (such as a specific printer driver), whereas application software instead specifically includes examples such as word processing programs (like Microsoft Word), popular web browsers (like Google Chrome), and various different computer games (like Minecraft); it is generally true that application software, considered as a broad overall category, tends to be considerably more varied, diverse, and numerous than system software, precisely because there exists such an enormously wide and continually expanding range of different specific tasks and activities that users might potentially wish to accomplish using their computers. Finally, in terms of how each of these two different categories of software is typically installed onto a computer system in the first place, system software is very generally pre-installed directly onto a computer by its original manufacturer before the computer is ever sold to an end user, since a computer genuinely cannot properly or reliably function at all without having at least some form of basic operating system already installed and actively running on it from the very moment it is first switched on; application software, however, is instead typically installed by the individual end user themselves, entirely according to their own specific personal needs, professional requirements, and individual personal preferences — for instance, a professional graphic designer working in the creative industry would likely choose to specifically install specialized graphic design software such as Adobe Photoshop onto their particular computer, whereas a different user working primarily within a more general business or office context might instead choose to specifically install word processing and spreadsheet software such as Microsoft Word and Microsoft Excel onto their own particular computer setup.

Discuss the main functions of commonly used application software, such as word processing, spreadsheet, and graphic design applications, with specific examples of each category.

Application software exists in a very wide, diverse, and continually growing range of different specific forms, each one specifically designed and purpose-built to help computer users accomplish a particular, well-defined type of task or activity,



and three of the most commonly used, widely recognized, and genuinely significant categories of application software worth specifically discussing in detail are word processing software, spreadsheet software, and graphic design software, each of which serves a distinctly different core function within the broader overall software ecosystem. Word processing software is very specifically designed for the core purposes of creating, subsequently editing, carefully formatting, and eventually printing text-based documents, and it therefore represents an absolutely essential category of software tool for tasks such as writing formal letters, drafting detailed professional reports, and composing academic essays or similar text-based assignments; well-known, widely used examples of word processing software specifically include Microsoft Word (which offers a very comprehensive range of built-in features, including text formatting tools, automatic spell-checking, and integrated grammar-checking functionality), Google Docs (a web-based word processor that is particularly notable for enabling convenient real-time collaboration, whereby multiple different users are able to simultaneously edit one single shared document together at the very same time), Apple Pages (a word processor specifically designed for Apple's own macOS and iOS operating systems), and LibreOffice Writer (a completely free, fully open-source software alternative that nevertheless still offers many of the very same core features and capabilities as the considerably more well-known Microsoft Word). Spreadsheet software, by contrast, is instead very specifically designed for the core purposes of organizing, subsequently analyzing, and reliably storing numerical and other forms of data within a clearly structured tabular grid format, made up of individual data cells that are themselves neatly arranged into a clear, well-organized system of distinct rows and columns, thereby making spreadsheet software an absolutely essential software tool for common tasks such as personal or household budgeting, detailed financial analysis, and broader statistical data analysis; well-known, widely used examples of spreadsheet software specifically include Microsoft Excel (which offers powerful, sophisticated built-in features such as complex mathematical formulas and highly flexible pivot tables for data summarization), Google Sheets (a web-based spreadsheet application that similarly supports convenient real-time collaborative editing among multiple different users), Apple Numbers (a spreadsheet application specifically designed for Apple's own macOS and iOS operating systems, particularly well known for its strong built-in data



visualization capabilities), and LibreOffice Calc (a completely free, fully open-source software alternative to the considerably more well-known Microsoft Excel). Graphic design software, finally, is instead very specifically designed for the different core purposes of creating, subsequently editing, and properly managing various forms of visual content, and it consequently represents an absolutely essential category of software tool for professional designers, practicing artists, and indeed anyone else who happens to be actively working with visual media of any kind, across a wide range of different creative and professional industries, including advertising, dedicated web design, publishing, and broader multimedia production; well-known, widely used examples of graphic design software specifically include Adobe Photoshop (a particularly powerful, industry-standard software tool specifically for professional photo editing and digital painting), Adobe Illustrator (a specialized vector graphics editor specifically used for creating scalable logos and other similar types of illustrations), and Canva (a considerably more beginner-friendly, fully web-based graphic design tool that conveniently provides a very wide range of ready-made templates, specifically intended to make the entire process of graphic design work considerably more genuinely accessible to complete beginners and non-specialists). Taken together as a complete, integrated whole, these three genuinely distinct categories of application software collectively serve to clearly illustrate the truly enormous overall breadth, diversity, and practical everyday usefulness of application software as a very broad general category, extending well beyond simple everyday text-based document creation, to additionally encompass sophisticated numerical data analysis and highly creative visual design work as well.

Multiple Choice Questions (MCQs)

What is the primary function of an operating system? (A) To create documents (B) To manage hardware resources and provide a user interface (C) To perform calculations (D) To design graphics

Correct answer: (B) To manage hardware resources and provide a user interface. The operating system's primary function is managing hardware resources and providing a user interface, along with running applications.



Which software is used to enhance system performance and security? (A) Operating system (B) Utility software (C) Application software (D) Device drivers

Correct answer: (B) Utility software. Utility software (like antivirus programs and disk cleanup tools) is specifically used to enhance system performance, security, and maintenance.

What role do device drivers play in a computer system? (A) Manage files (B) Facilitate communication between hardware devices and the operating system (C) Create presentations (D) Enhance graphics performance

Correct answer: (B) Facilitate communication between hardware devices and the operating system. Device drivers act as translators, facilitating communication between hardware devices and the operating system.

Which of the following is an example of application software? (A) Microsoft Word (B) BIOS (C) Disk Cleanup (D) Device Manager

Correct answer: (A) Microsoft Word. Microsoft Word is application software, designed to help users perform the specific task of word processing; the others are system software.

What is the main purpose of spreadsheet software? (A) To edit text documents (B) To organize and analyze data (C) To create visual content (D) To enhance system security

Correct answer: (B) To organize and analyze data. Spreadsheet software's main purpose is to organize, analyze, and store data in a tabular grid of rows and columns.

How does utility software differ from application software? (A) Utility software manages hardware, while application software performs specific tasks for users. (B) Utility software creates documents, while application software manages hardware. (C) Utility software performs specific tasks for users, while application software manages hardware. (D) Utility software is free, while application software is paid.

Correct answer: (A) Utility software manages hardware, while application software performs specific tasks for users.. Utility software (part of system software) helps manage and maintain hardware/system performance, while application software performs specific tasks for the user.



Which type of software would you use to design a logo? (A) Operating system (B) Spreadsheet software (C) Graphic design software (D) Utility software

Correct answer: (C) Graphic design software. Graphic design software (such as Adobe Illustrator) is specifically used for creating logos and other scalable graphics.

What is the function of system software? (A) To facilitate communication between hardware and software (B) To perform specific tasks for the user (C) To create visual content (D) To organize and analyze data

Correct answer: (A) To facilitate communication between hardware and software. System software's core function is to manage hardware and facilitate communication between hardware and other software, including applications.

Why are operating system updates important? (A) They increase screen brightness (B) They add more fonts (C) They enhance security and fix bugs (D) They improve battery life

Correct answer: (C) They enhance security and fix bugs. OS updates are important primarily because they enhance security and fix bugs, protecting the system against newly discovered vulnerabilities.

What is a common task you can perform using word processing software? (A) Create and edit text documents (B) Manage hardware resources (C) Enhance system performance (D) Organize and analyze data

Correct answer: (A) Create and edit text documents. Word processing software is used to create, edit, format, and print text-based documents.

Quick Revision Summary

- Software = System Software (manages hardware, e.g. OS, drivers, utilities) + Application Software (performs user tasks, e.g. Word, browsers, games)
- 5 major OSes: Windows, macOS, Linux, Android, iOS
- OS has 3 main functions: manage hardware resources, provide user interface (GUI/CLI), run applications



- GUI = visual (icons, windows, menus, beginner-friendly); CLI = text commands (flexible, powerful, harder for beginners)
- Utility programs: Disk Cleanup, Antivirus, Backup Software, File Compression Tools
- Device drivers = translators between hardware and OS; work in 3 stages: Installation → Communication → Operation
- Word processors: MS Word, Google Docs, Apple Pages, LibreOffice Writer
- Spreadsheets: Excel, Google Sheets, Apple Numbers, LibreOffice Calc; Graphic design: Photoshop, Illustrator, CorelDRAW, GIMP, Canva

Exam Tips

- Remember the 3 core OS functions in order: manage hardware resources → provide user interface → run applications
- GUI vs CLI: GUI = visual/beginner-friendly; CLI = text-based/powerful — a frequently tested distinction
- System software is usually pre-installed; application software is usually installed by the user — a key differentiator often tested
- Memorize at least 2 examples for each utility program type (disk cleanup, antivirus, backup, file compression)
- The device driver 'TV remote control' analogy (TV=device, remote=driver, you=computer) is a great way to explain and remember this concept
- For application software questions, always be ready to name at least 2 examples per category: word processing, spreadsheet, graphic design

