

# Introduction to Operating System

An operating system acts as an intermediary between the computer hardware and the user. It prevents the user from directly accessing the hardware and manages all system resources.

- It provide an environment in which a user can execute programs conveniently and efficiently.
- Operating system (OS) is a program that runs at all times on a computer.
- It assigns resources such as memory, processors, and input/output devices to processes that need them.

## Operating System and User Interaction

Every general-purpose computer consists of hardware, an operating system, system programs and application programs.

- The hardware includes CPU, ALU, memory, I/O devices, peripheral devices and storage devices.
- The system programs include compilers, loaders, editors and the OS itself.
- The application programs are user-level software that perform specific tasks.

An operating system is always running in the background to manage and coordinate the use of hardware resources among system and application programs for multiple users. The OS itself is not only a user interface. Instead, it provides an interface for user interaction, typically through:

- Command-Line Interface (CLI), (e.g., Bash, PowerShell)
- Graphical User Interface (GUI), (e.g., Windows desktop, macOS Finder)

**Note:** At the core of the OS lies the Kernel, which is the primary interface between hardware and software which handles low-level operations such as process management, memory management, file system control and device handling.

## Goals of Operating System

An operating system is system software that acts as an interface between the user and computer hardware. It manages system resources and provides a platform for executing programs efficiently.

### Primary Goals

The primary goals of an operating system (OS) are to provide an easy to use and convenient environment for executing user programs.

- **User Convenience:** It should be easy to use, providing a user-friendly interface and making it simple to interact with the system.
- **Program Execution:** It facilitates the execution of user programs, providing the necessary environment and services for them to run.
- **Resource Management:** The OS manages and allocates the computer's resources, including the CPU, memory, disk storage and input/output devices, to ensure fair utilization.
- **Security:** The OS protects the system and user data from unauthorized access, ensuring confidentiality, integrity and availability of information.

### Secondary Goals

Secondary goals of an operating system focus on improving overall system performance and stability. They ensure efficient use of resources and make the system reliable and robust.

- **Efficient Resource Utilization:** It should aim to maximize the performance and utilization of computer resources like CPU, Memory and I/O devices, ensuring that the system runs smoothly and efficiently.
- **Reliability:** It should be robust and reliable, able to handle errors and exceptions gracefully, ensuring that the system continues to operate smoothly. It should be modular in design and easy to debug.

## Components of an Operating System

There are two basic components of an Operating System.

- Shell is the outermost layer of the Operating System and handles user interaction. It interprets input for the OS and handles the output from the OS.
- Kernel is the core component of the operating system. The kernel is the primary interface between the Operating system and Hardware.

## List of Common Operating Systems

There are multiple types of operating systems each having its own unique features:

- **Windows OS :** Developed by Microsoft. Used in personal computing, business environment, gaming.
- **macOS :** Developed by Apple. Used for creative industries (design, video editing, music production), Personal computing and Professional environments.
- **Linux :** Developed by Linus Torvalds and maintained by the open-source community (Linux Foundation, Red Hat, Canonical, etc.). Used in servers and data centers, development and programming, Personal computing for tech enthusiasts.
- **Unix :** Developed originally at AT&T Bell Labs, various commercial and open-source versions available. Used in servers and workstations, Development environments, Research and academic settings

# Applications of Operating System

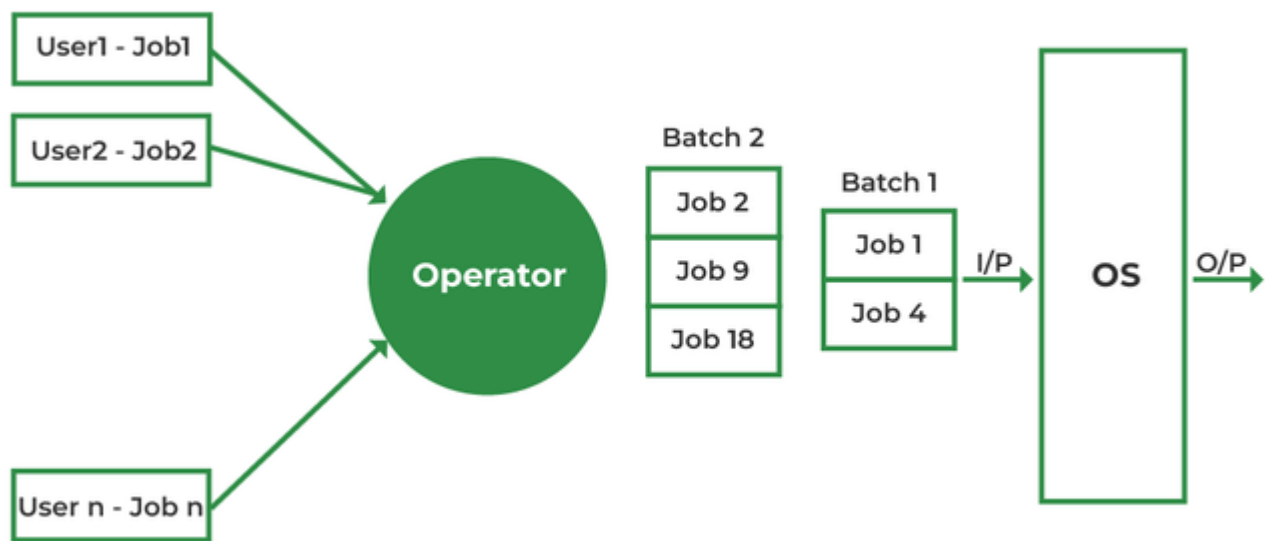
- **Platform for Application Programs:** It provides a platform, on top of which, other programs, called application programs can run.
- **Managing Input-Output Unit:** It also allows the computer to manage its own resources such as memory, monitor, keyboard, printer, etc. Management of these resources is required for effective and fair utilization.
- **Multitasking:** It manages memory and allows multiple programs to run in their own space and even communicate with each other through shared memory.
- **Manages Memory and Files:** It manages the computer's main memory and secondary storage. Additionally, it allows and deallocates memory to all tasks and applications.
- **Provides Security:** It helps to maintain the system and applications safe through the authorization process. Thus, the OS provides security to the system.

## Types of Operating Systems

An operating system (OS) is software that manages computer hardware and software resources. It acts as a bridge between users and the computer, ensuring smooth operation. Different types of OS serve different needs; some handle one task at a time, while others manage multiple users or real-time processes.

### 1. Batch operating System

A [Batch Operating System](#) is designed to handle large groups of similar jobs efficiently. It does not interact with the computer directly but instead processes jobs that are grouped by an operator. These jobs are queued and executed one after the other, without user interaction during the process.



*Executes a group of similar jobs automatically in batches without user interaction.*

## Advantages

- **Minimal Idle Time:** The system minimizes idle time by processing jobs in a continuous sequence without human intervention.
- **Handling Repetitive Tasks:** Ideal for managing large, repetitive tasks, such as payroll and billing, with minimal effort.
- **Improved Throughput:** Batch systems can handle high volumes of jobs at once, improving overall system throughput.

## Disadvantages

- **Inefficient CPU Utilization:** When a job is waiting for input/output (I/O), the CPU remains idle, leading to poor utilization of resources.
- **Increased Response Time:** The time between job submission and output can be high as all jobs are processed sequentially.
- **Lack of Real-Time Feedback:** Users cannot interact with the system in real-time, making it less suitable for interactive tasks.

**Examples:**

- *Insurance Claim Processing*
- *Library Book Records*
- *Stock Market Reports*

## 2. Multi-Programming Operating System

In a Multi-Programming Operating System, multiple programs run in memory at the same time. The CPU switches between programs, utilizing its resources more effectively and improving overall system performance.

### Advantages

- **Better CPU Utilization:** CPU stays busy by switching to another job during I/O wait.
- **Improved Throughput:** Multiple jobs run concurrently, increasing work done per unit time.
- **Efficient Resource Use:** CPU, memory, and I/O devices are shared effectively among processes.

### Disadvantages

- **Complex Design:** Requires advanced memory management and CPU scheduling.
- **Security Issues:** More programs in memory increase chances of unauthorized access.
- **High Memory Requirement:** Needs larger RAM to run multiple programs together.

### Examples:

- *Banking systems*
- *Railway servers*
- *Billing machines*

### 3. Multi-tasking/Time-sharing Operating systems

Multitasking OS is a type of multiprogramming system where each process runs in a round-robin manner. Every task gets a fixed time slice called a quantum. After the quantum ends, the OS switches to the next task, allowing multiple tasks whether from one user or many to run smoothly on a single system.

#### Advantages

- **Equal CPU Access:** Each task gets a fair share of CPU time.
- **Reduced Software Duplication:** Many users can run the same software without needing separate copies.
- **Low CPU Idle Time:** Efficient scheduling keeps the CPU busy.

#### Disadvantages

- **Lower Reliability:** System failures affect all users.
- **Security Concerns:** Multiple users increase risks to data integrity and privacy.
- **Communication Issues:** Data sharing between users can cause conflicts.

#### Examples:

- *IBM VM/CMS*
- *TSO (Time Sharing Option)*
- *Windows Terminal Services*

## 4. Multi-Processing Operating System

Multi-Processing OS is a type of Operating System in which more than one CPU is used for the execution of processes. It better the throughput of the System.

### Advantages

- **Faster Processing:** Multiple CPUs work simultaneously, increasing overall system speed.
- **High Reliability:** If one processor fails, others can continue working (fault tolerance).
- **Supports Heavy Tasks:** Ideal for computation-intensive applications like scientific or industrial tasks.

### Disadvantages

- **High Cost:** Multiple processors and complex hardware increase system cost.
- **Complex Design:** Requires advanced OS support for communication and task distribution.
- **Not Always Efficient:** Poor task distribution can lead to idle processors and wasted resources.

### Examples:

- *UNIX*
- *Linux (Ubuntu, Red Hat, Debian)*
- *macOS*

## 5. Distributed Operating System

Distributed operating systems connects multiple independent computers through a shared communication network. Each system has its own CPU

and memory but works together as a single unit. The main benefit is remote access, allowing users to use files and software stored on other connected systems.

## Advantages

- **Independent Systems:** Failure of one machine does not affect others.
- **Easily Scalable:** New systems can be added to the network easily.
- **Lower Processing Delays:** Tasks are handled faster across multiple machines.

## Disadvantages

- **Network Dependency:** If the main network fails, communication stops.
- **Lack of Standardization:** No well-defined language or model for building such systems.
- **High Cost & Complexity:** Hardware is expensive, and the software is complex and not widely understood.

### Examples:

- *LOCUS*
- *MICROS*
- *Amoeba*

## 6. Network Operating System

A Network Operating System (NOS) runs on a server and manages data, users, security, applications, and other network functions. It allows shared access to files, printers, and resources within a small private network. Users can see the configuration and connections of other users, which is why these systems are considered tightly coupled systems.

## Advantages

- **Centralized and Stable Servers:** Provide reliable management of resources.
- **Easy Upgrades:** New hardware and technologies can be added without difficulty.
- **Remote Access:** Users can access the server from different locations and devices.

## Disadvantages

- **High Server Cost:** Setting up and maintaining servers is expensive.
- **Dependency on Server:** Most operations rely on a central server.
- **Regular Maintenance Needed:** Frequent updates and technical support are required.

### Examples:

- *Microsoft Windows Server 2003*
- *UNIX, Linux*
- *Mac OS X*

## 7. Real-Time Operating System

These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called **response time**. **Real-time systems** are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

### Types of Real-Time Operating Systems

- **Hard Real-Time Operating System:** Used where strict timing is essential and any delay is unacceptable, such as airbags or automatic

parachutes. These systems avoid virtual memory to ensure immediate response.

- **Soft Real-Time Operating System:** Used where timing is important but minor delays are acceptable. These systems aim to give quick and predictable responses but do not require perfect accuracy. They are commonly used in multimedia applications, gaming, video streaming, and other interactive tasks.

### Hard Real-Time OS and Soft Real-Time OS.

## Advantages

- **Maximum Utilization:** Maximum utilization of devices and systems, thus more output from all the resources.
- **Error-Free:** These types of systems are error-free.
- **Memory Allocation:** Memory allocation is best managed in these types of systems.

## Disadvantages

- **Limited Tasks:** Very few tasks run at the same time and they focus on very few applications to avoid errors.
- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write.
- **Thread Priority:** It is not good to set thread priority, as these systems are not suitable for frequent task switching.

## Examples:

- *Scientific experiments*
- *Medical imaging systems*
- *Robots*

## 8. Mobile Operating Systems

Mobile operating systems are designed specifically for mobile devices such as smartphones and tablets. Examples of such operating systems are Android and iOS. These operating systems manage the hardware and software resources of the device, providing a platform for running applications and ensuring a seamless user experience.

### Advantages

- **User-Friendly Interfaces:** Mobile operating systems are designed to be intuitive and easy to use, making them accessible to a wide range of users.
- **Extensive App Ecosystems:** The availability of a vast number of applications allows users to customize their devices to meet their specific needs.
- **Connectivity Options:** Mobile operating systems support multiple connectivity options, enabling users to stay connected wherever they go.

### Disadvantages

- **Battery Life Constraints:** Despite advancements in power management, battery life remains a challenge for mobile devices, especially with heavy usage.
- **Security Risks:** Mobile devices are susceptible to various security threats, such as malware and phishing attacks, which can compromise user data.
- **Fragmentation:** In the case of Android, the wide range of devices and customizations can lead to fragmentation, making it difficult for developers to ensure compatibility across all devices.

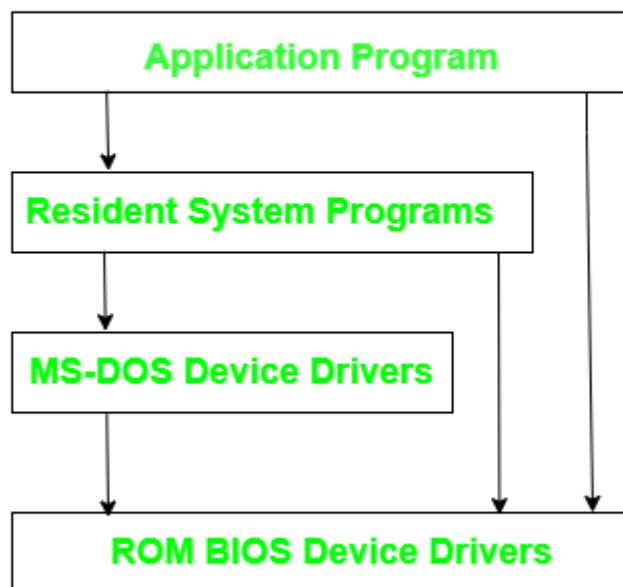
**Examples:**

- *Android*
- *iOS*
- *Blackberry*

## What is MS-DOS Operating System?

MS-DOS Operating System also called the Disk Operating system was Developed by Microsoft for x86 personal computers. It works on the phenomenon of doing less and getting more. It is a 16-bit operating system. A closed-source model was initially released on August 12, 1981, and the final release on September 14, 2000.

### MS-DOS layer Structure



# Why is MS-DOS Operating System Used?

It is used for many purposes:

- **Education and research:** It is used in the Education and research field to know how simple MS-DOS is and how it works. The source code is available from Microsoft in two versions.
- **Legacy systems:** Some organizations prefer using MS-DOS because it has been used reliably over decades whereas others don't have the budget to upgrade the application since this process is costly.
- **Embedded systems:** [Embedded systems](#) are dependent on MS-DOS especially new devices with x86 processor architecture. It is also used on legacy Embedded systems which have been used for decades.
- **Classic Computer Games:** There are still some games that were developed on MS-DOS and still support only MS-DOS.

## Features of MS-DOS Operating System

- It is a minimalist OS which means it can boot a computer and run programs.
- Still usable for simple tasks like word processing and playing games.
- The mouse cannot be used to give inputs instead it uses basic system commands to perform the task.
- It is a 16-bit, free [operating system](#).
- It is a single-user operating system.
- It is very lightweight due to fewer features available and no multitasking.

## Working of MS-DOS

MS-DOS input and output occur at the command line in the form of issued commands and system results of those commands. When booted, MS-DOS systems display a command prompt. MS-DOS commands are then entered using a keyboard. Commands are either one of the standard DOS

commands, implemented as part of the OS, or they are the file name of an executable program.

Although it does not natively support a GUI, MS-DOS can run graphical programs if they are installed on a system disk or diskette. Users enter the command for a graphical program, and the program takes control of the input and output devices. When the program terminates, the control reverts to MS-DOS and the command prompt reappears.

## Limitations of MS DOS Operating System

- It is not a multitasking operating system that is we cannot run too many applications in the background.
- Files on the system can be easily deleted or the system can also be easily destroyed.
- It does not provide any warning message before you delete or perform any unwanted task like in [windows or Linux](#).
- It is text-based and it does not have any graphical user interface.
- Not secure to be used in any kind of public network.
- [Encryption](#) is not supported.
- Difficulty in memory access.
- Mouse cannot be used to give inputs.

## Types of MS-DOS Commands

There are mainly two types of MS-DOS commands:

### 1. Internal Commands

Internal commands are those commands that are loaded automatically in the memory when DOS is loaded into memory during the booting process. These commands are easier to learn and use. They require no external files for their storage as in the case of external commands. These are for performing a basic operation on files and in directories. They do not need

any external file support. These commands are used for common jobs such as copying and erasing files.

## **2. External Commands**

These external commands are for performing advanced tasks and they do not need some external file support as they are not stored in COMMAND.com. The external commands are used less frequently and are stored in some external files which are stored in some secondary storage devices. Whenever an external command is to be executed then the external file in which that particular command is stored is transferred from the secondary storage disk to the main memory(RAM).

## **Basics of Batch Scripting**

Batch Scripting consists of a series of commands to be executed by the command-line interpreter, stored in a plain text file. It is not commonly used as a programming language and so it is not commonly practiced and is not trending but its control and dominance in the Windows environment can never be neglected. Almost every task and every action can be performed and executed by a simple sequence of commands typed on the Windows Command Prompt.

### **Batch Script execution**

There are 2 ways to execute a batch script.

- Type the batch script in the command prompt.
- Write the code of script in a file and execute it through the command prompt.

Typing commands again and again on the terminal can be a very tedious task to do if we have a very lengthy code. So option 2 is generally preferred to create batch files.

# Creating Batch Files

Steps to create a Batch file are pretty simple:-

1. Create a new text file with a **'.txt'** extension.
2. Now rename this file with extension as **'.bat'** this creates a Batch file.
3. Now open this .bat file in any text editor and start scripting.

To begin scripting we must be aware of the commands of the batch interface. The commands of Batch are sometimes similar to Linux Scripting commands.

## Batch Commands

Basic batch commands are all **case insensitive** and can be used to perform a specific set of instructions:-

- **DIR** - The 'dir' command is used to get all the directories, sub-directories, and files present in the current working directory.
- **CD** - The 'cd' command is used to change the current working directory.
- **VER** - The 'ver' command tells the version of the user's Windows.
- **CLS** - The 'cls' command is used to clear the screen of the command prompt.
- **ECHO** - The 'echo' command is by default 'on' but if we turn it off by 'echo off' it turns off prompt till the time 'echo on' is passed.
- **@** - The '@' if used before any command hides which command is running.
- **@ECHO OFF** - This commands serves as the start point to any basic batch script as it hides the prompt with 'echo off' and hides 'echo off' command with '@'.
- **HELP** - This command tells us all about the commands available in the cmd. It runs only if the cmd is run as an administrator.

## **Unit 4 – Windows operating environment**

### **Microsoft Windows Features**

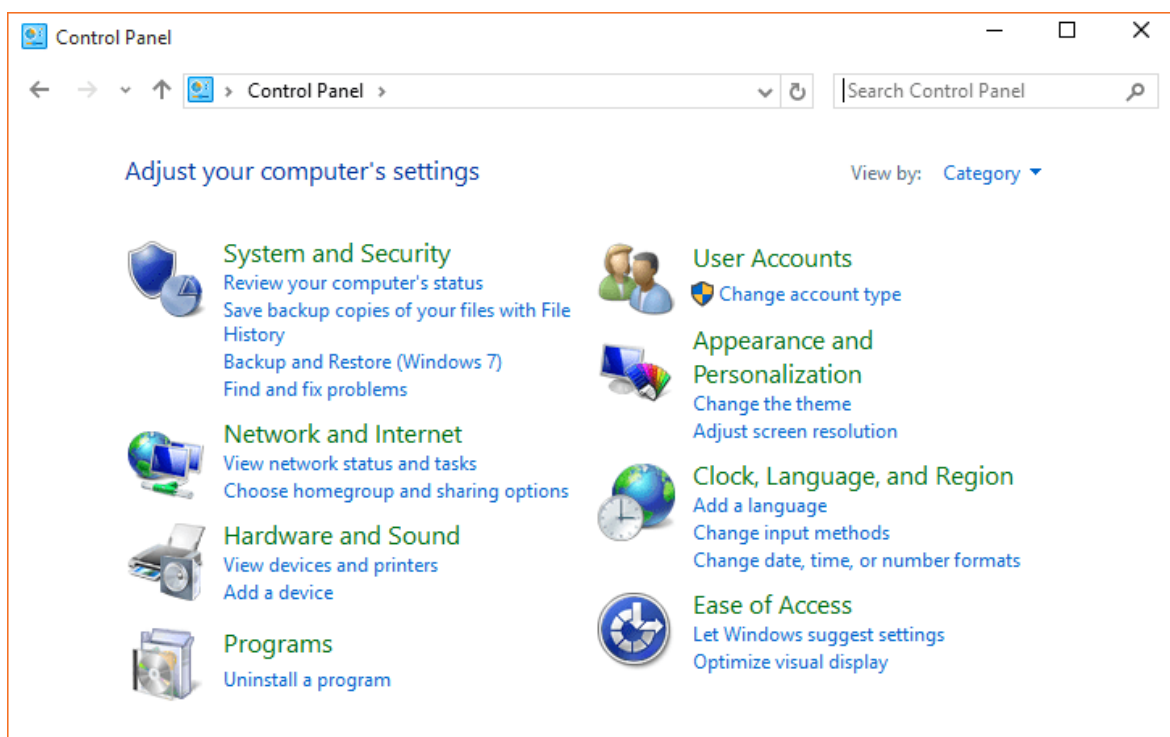
Microsoft Windows is an operating system created by Microsoft for PC computers. An operating system is the software utilized to run and manage programs and functions on the computer. Windows updates and adds to its features with each release,

however some of the features it is known for are Windows Live, Windows Search, Windows Updates and Windows Taskbar.

## Control Panel

The **Control Panel** is a section of Microsoft Windows that enables a user to change various computer hardware and software features. Settings for the mouse, display, sound, network, and keyboard represent a few examples of what may be modified in the Control Panel. Below are some examples of how the Control Panel appeared in Windows.

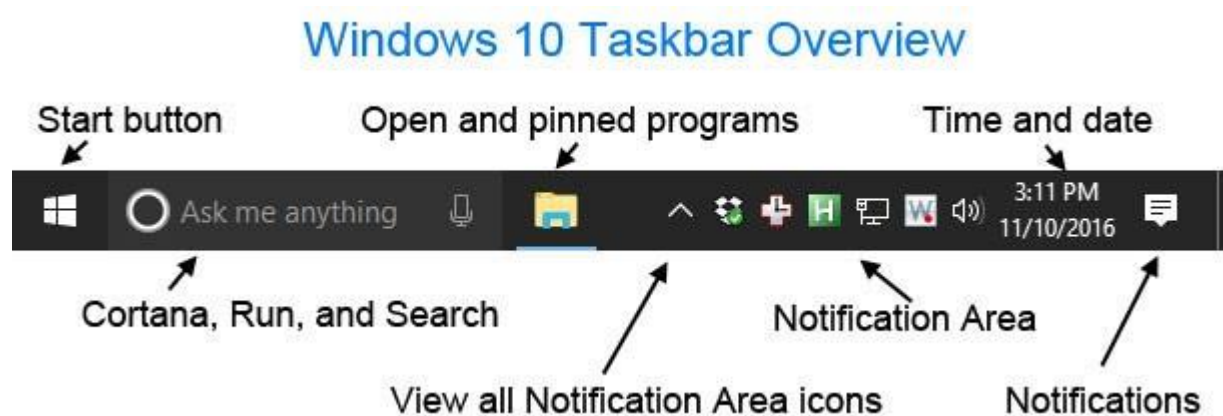
### The evolution of the Windows Control Panel



## Taskbar

The **taskbar** is an element of an operating system located at the bottom of the screen. It allows you to locate and launch programs

through Start and the Start menu, or view any program that is currently open. The taskbar first introduced with Microsoft Windows 95 and can be found in all subsequent versions of Windows. It also allows them to check the date and time, items running in the background through the Notification Area, and with early versions of Windows access to the Quick Launch.



## Desktop

A **desktop** may refer to any of the following:

1. A **desktop** is a term commonly used to describe a desktop computer or system unit.
2. When referring to an operating system or GUI (graphical user interface), the **desktop** is a system of organization of icons on a screen. The Microsoft Windows desktop was first introduced with Microsoft Windows 95 and included with every version of Windows since then. The image below is an example of the Microsoft Window

## Windows 7 Desktop



## Windows Application

Windows Application is a user build an application that can run on a Windows platform. The windows application has a graphical user interface that is provided by Windows Forms. Windows forms provide a variety of controls including Button, TextBox, Radio Button, CheckBox, and other data and connection controls. You can easily design a web application using an IDE Microsoft Visual Studio using a variety of languages including C#, Visual Basic, C++, J# and many more

# Icon

Windows XP



Windows 7



Windows 8



Windows 10



An **icon** is a small graphical representation of a program or file. When you double-click an icon, the associated file or program will be opened. For example, if you were to double-click on the My Computer icon, it would open Windows Explorer. Icons are a component of GUI operating systems, including Apple macOS X and Microsoft Windows. Icons help users quickly identify the type of file represented by the icon. The image to the right is an example of “My Computer” icons in different versions of Microsoft Windows.

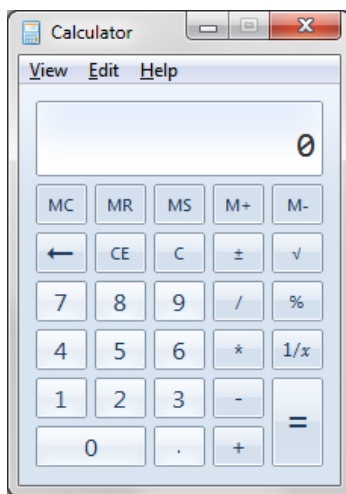
## Windows Accessories – Operating Systems

Windows operating system ships with some handy applications known as Windows accessories. Calculator, Notepad, Paint,

Explorer, WordPad are some of the most frequently used accessories.

Apart from above mentioned applications, Windows has a few tools for Ease of Access and some System Tools. We'll be briefly talking about them here.

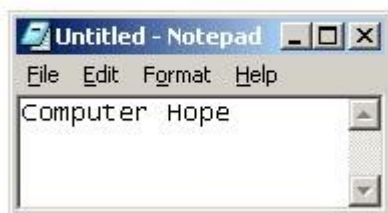
## Calculator



Calculator is a calculating application included in all the versions of Windows. It can be used to perform simple calculation, scientific calculation and Programming calculation.

## Notepad

Microsoft Notepad text editor



**Notepad** is a generic text editor included with all versions of Microsoft Windows that allows you to create, open, and read plaintext files. If the file contains special formatting or is not a plaintext file, it will not be able to be read in Microsoft Notepad. The image to the right is a small example of what the Microsoft Notepad may look like while running.

**Paintbrush** is a raster image editor for Mac OS X. It aims to replace MacPaint, an image editor for the classic Mac OS last released in 1988. It also is an alternative to MS Paint. It has basic raster image editing capabilities and a simple interface designed for ease of use. It exports as PNG, JPG, BMP, GIF, and TIFF. The application also is often used for pixel art because of its grid option, and is not made for large scale images or GIMP or Photoshop-like editing on pictures or photographs.