UNIT 1

Operating Systems: Structure and Services

Student Learning Outcomes:

By the end of this chapter, you will be able to:

- Define an operating system and explain its role as an interface between the user and translate user actions into hardware-level instructions.
- Identify the responsibilities of the OS in managing multiple user accounts.
- Differentiate between the kernel and shell, including their specific functions.
- Explain OS layers and how each layer interacts with the others.
- Identify the role of system libraries and device drivers in OS functionality.
- Explain the stages in the process lifecycle (creation, execution, termination) with examples and apply the First-Come, First-Served (FCFS) process scheduling.
- Describe multitasking and concurrency, giving one real-world analogy.
- Distinguish between primary memory (RAM) and virtual memory.
- Define a process and a thread, and explain how threads share resources within a process.
- Give examples of how multithreading improves performance in applications.
- Define a system call and explain its purpose and types
- Define file system, files, folders, and metadata, and explain their roles in organizing data.
- Explain how these components work together to process a user request from device to server and back.

Introduction

In this chapter, we will discuss the core functions and structure of an operating system, the fundamental software that manages computer hardware and facilitates seamless user interaction. The discussion will covers:

- operating system architecture and its role as the central controller of the computer; process management concepts such as the
- process lifecycle, multitasking, concurrency, and scheduling
- memory management, multithreading, and the use of system calls to securely access hardware resources

- file system organization and the main types of operating systems, including real time, embedded, network, and mobile OS.

1.1 Introduction to Operating System (OS)

Operating System (OS) is a type of system software that manages hardware, runs application software, and provides a user interface like Windows, macOS, Linux, Android, and iOS, along with their basic roles. Here we will go a step further and focus on how the operating system works as the central controller of the computer system, ensuring smooth and secure interaction between the user and the hardware, especially in multi-user environments.

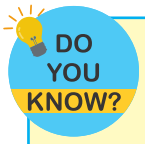
1.1.1 Operating System (OS) as the Central System Controller

An operating system works like a traffic controller for the computer. It decides which task should be done first, how the computer's memory is used, and which devices (like a printer or speakers) should be active at a given time. In short, we can say that the OS makes sure the computer works smoothly even when multiple tasks are running at the same time.

1.1.2 Role in User–Hardware Interaction

Computer hardware cannot understand human language directly. The operating system acts as a translator between the user and the hardware system.

Example: When you click an icon or type something, the OS changes those actions into instructions that the hardware can understand. This allows people to use computers easily without learning how the hardware works.



Humans speak languages like Urdu or English, whereas computers understand only binary language, or ON and Off switch i.e. 0 and 1.

Example: When you press the letter **A** on a keyboard, the operating system converts it into a special binary code:

- The letter **A** in binary is $(\mathbf{01000001})_2$.
- This code is then sent to the computer screen, which displays **A** on the screen.
- This process happens so quickly that it feels instant to the user.

1.1.3 Responsibilities in Multi-User Environments Tasks

In places like schools, offices, or online systems, many people may use the same computer. The operating system ensures:

- The creation and management of separate **user accounts**.
- Keeps each user's files and information **private**.
- Shares the computer's resources fairly so that no one slows down the systems functionalities.
- Protect shared resources from **unauthorized access**.

1.1.4 Creating and Managing User Accounts

The operating system allows separate accounts for different users, each with its own desktop, files, settings, and passwords.

Example: In computer lab, students login with a username and password provided by the school. This keeps each student's work separate and private. In most modern OS like Windows, a new account can be created through the Settings or Control Panel by selecting User Accounts, choosing Add New User, and entering details like username, password, and account type(Standard, Administrative, Guest).

1.2 Architecture of an Operating System

The architecture of an operating system is the way how its parts are organized and how they work together. Each part has a special role, and together they make the computer work smoothly just like a school has different departments that perform specific duties but work together for the smooth working of the school.

1.2.1 Kernel vs Shell

I. **Kernel:** The kernel is the core part of the operating system. That directly controls the computer's system software and hardware such as the CPU, memory, and devices, as shown in Figure 1.1. It decides how and when different programs can use these resources.
Example: When you open a file, the kernel manages the process of reading it from the hard drive and sending it to the screen. Like Engine of a car work is kernel and accessories like steering wheel, dashboard is shell.

II. **Shell:** The shell is the outer part of the OS that interacts with the user, also depicted in Figure 1.1. It receives commands from the user and passes them to the kernel. Shells can be:
Graphical shells (like Windows desktop) where you click icons and use menus.
Command-line shells (like the Command Prompt or Terminal) are where you type instructions.

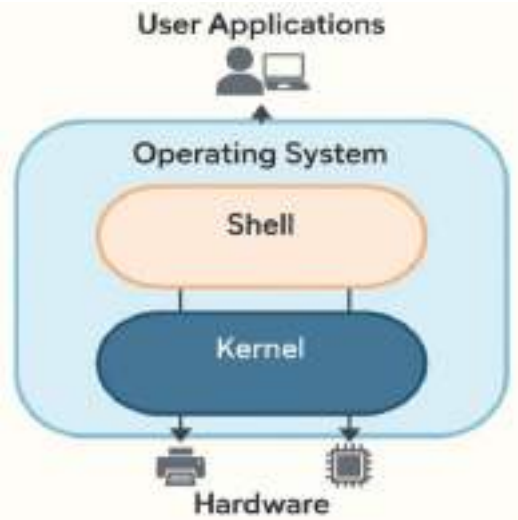


Figure 1.1: Diagram to illustrate the relationship between the user, shell, and kernel in an operating system.

1.2.2 OS Layers and Modular Design

In operating systems, the design is divided into **layers**, where each layer has a specific job:

- **Lower layer** work directly with hardware devices like the CPU, RAM, and storage, Hard disk.
- **Middle layer** manage these resources and make sure programs can use them when needed.
- **Upper layer** run applications and provide the interface that the user views on the screen.

Each layer depends on the one below it. This design makes the operating system easier to manage, repair, and improve without changing the whole system.

Example: A school system can be used as an example to understand the layered architecture of an operating system:

- The **support staff** (like guards and cleaners) work at the base, keeping the school ready (like the lower layers working with hardware).
- The **administration** manages resources, schedules, and rules (like the middle layers managing memory and storage).
- The **teachers and students** use these arrangements to teach and learn (like the upper layers running applications and interacting with users).

1.2.3 System Libraries and Device Drivers

System libraries are collections of ready-made instructions that programs can use to perform common tasks, such as opening files or showing text on the screen.

Example: When a photo editing app needs to open an image, it uses the operating system's library to read the file. This way, the app does not have to create its own method to open pictures.

Device drivers are special programs that allow the operating system to communicate with hardware devices such as printers, keyboards, and graphics cards.

ACTIVITY

Objective:

Identify the **shell, device driver,** and **system library** used by your computer.

Steps:

1. Locate the shell on your computer (desktop or command-line).
2. Open Device Manager (Windows) or System Information (Mac/Linux).
3. Find one hardware device (e.g., printer, keyboard) and note its driver name.
4. Open a simple program (e.g., Calculator) and identify one task it performs that may use a system library (e.g., displaying numbers).
5. Record your findings in a table with these columns:
    - **OS Component** (Shell / Device Driver / System Library)
    - **Your Example** (e.g., Windows desktop, HP printer driver, Calculator app)
    - **Purpose/Role** (what it does in the system).



1.3 Process Management in Operating System (OS)

The operating system is responsible for managing all the programs that run on a computer. In this context, these running programs are called **processes**. Process management ensures that each process gets the **resources** it needs, even when multiple processes are active. Process management is one of the most important jobs of an operating system because it ensures all processes run smoothly without disturbing each other.

1.3.1 Process Life Cycle

A process goes through several stages during its life cycle:

1. **Creation:**
    - This happens when you start any program (like MS Word).
    - The OS loads the program into memory and gives it the resources (like CPU time and memory) it needs.
2. **Execution:**
    - The process is actively running and performing tasks effectively.
3. **Termination:**
    - The process finishes its task and is closed by the user or the system.
    - The OS frees the resources so they can be used by other processes.

Example: Process Lifecycle of opening a Web Browser on your mobile or computer like Google Chrome.

1. **Creation:** Begins when the user clicks the browser icon. The operating system loads the program into memory and allocates the required resources.
2. **Execution:** The browser performs tasks such as loading web pages, displaying media, and responding to user actions.
3. **Termination:** Occurs when the browser is closed. The operating system stops the process and releases its resources for other uses.

1.3.2 Multitasking and Concurrency

Modern operating systems can manage many processes so efficiently that it appears they are all running at the same time.

- **Multitasking:** The operating system allows more than one program to be open and usable by a single user at the same time. You can easily switch between them whenever you need.
  Example: You can listen to music, keep a document open, and browse the internet, moving between them as needed.
- **Concurrency:** More than one process is active at the same time in an OS, but the CPU processes them one by one in extremely fast cycles.
  Example: Like a chef preparing three dishes, working on one for a short time, then moving to the next, and repeating, the CPU switches between processes so rapidly that the user does not notice any delay.

1.3.3 Process Scheduling Concepts

We learned that many processes can be in progress during the same time period, but the CPU works on them one at a time in very fast turns. Since the CPU cannot run all processes at the same time, the operating system must decide:

- Which process should run first?
- How long each process should run.

This process is called **scheduling**. There are different scheduling methods used by operating systems. But in this chapter, we will explore only one of the simplest scheduling techniques, ***First Come, First Served (FCFS)***.

In the ***FCFS*** method, the CPU processes tasks in the exact order they arrive. The first process to arrive is completed first, and the next process starts only after the previous one finishes.

Example: Like a queue at a shop counter, the first customer in line is served first, then the next, and so on.

Numerical Example: Let us use some simple data to see how **FCFS** scheduling works in practice.

Process ID	Arrival Time	Time Needed
P1	0 seconds	5 seconds
P2	1 second	3 seconds
P3	2 seconds	2 seconds

Step-by-Step Explanation:

1. **At 0 seconds:**
   - **P1** arrives first, so it starts running immediately.
   - It requires **5 seconds** to finish.
   - It runs from **0s to 5s**.
2. **At 1 second:**
   - **P2** arrives while P1 is still running, so **P2** waits.
3. **At 2 seconds:**
   - **P3** arrives, but P1 is still running and P2 is already waiting.
   - In FCFS, P3 will also wait until both P1 and P2 have completed.
4. **At 5 seconds:**
   - P1 finishes.
   - The CPU now starts **P2**, which needs **3 seconds** to complete.
   - P2 runs from **5s to 8s**.
5. **At 8 seconds:**
   - P2 finishes.
   - The CPU now starts **P3**, which needs **2 seconds**.
   - P3 starts and runs from **8s to 10s**.

Execution Timeline:

[P1: 0–5] → [P2: 5–8] → [P3: 8–10]

```
0           5           8           10
|    P1     |    P2     |    P3     |
```

P1 : 0s – 5s (finish at 5)
P2 : 5s – 8s (finish at 8)
P3 : 8s – 10s (finish at 10)

Observation:

Even though **P3** has the shortest duration (only 2 seconds), it must still wait until **P1** and **P2** have finished because they arrived earlier. This is one of the main characteristics of FCFS; shorter tasks can be delayed if they arrive after longer ones.

Advantages and Disadvantages of FCFS:

Although FCFS is a simple and fair scheduling method, it has both advantages and disadvantages:

Advantages: Easy to understand and implement, as processes are served in the exact order they arrive. No process is skipped.

Disadvantages: Short processes may have to wait a long time if they are queued behind longer processes (known as the "convoy effect"), which can affect the overall efficiency of the system.

DO YOU KNOW?

The CPU in your computer can switch between processes **millions of times** in just one second. This happens so fast that you can play music, browse the web, and download files all at once.

ACTIVITY

Objective:

Use the **First Come, First Served (FCFS)** method to determine the order and completion time of processes.

Steps:

1. Consider the following processes:

Process	Arrival Time	Time Needed
P1	0s	4s
P2	1s	5s
P3	2s	3s

2. Arrange the processes in the order they will run according to the FCFS method.
3. Calculate the **start** and **finish time** for each process.

Answer the questions:

- What is the total time taken for all processes to finish?
- Which process had to wait the longest?

1.4 Memory

Memory is a fundamental component of a computer system, used to store data and instructions required for processing. It ensures that the CPU can access information quickly during program execution.

1.4.1 Primary Memory (RAM)

Primary memory, also called Random Access Memory (RAM), is the main working area of a computer. It is a fast storage area where the computer keeps

the data and instructions temporarily. Its data is erased when the computer is turned off. **Example:** When you open a Word document, the computer quickly places the program and the document into RAM so the CPU can work on them right away.



1.4.2 Virtual Memory

When the RAM is full, the operating system uses part of the computer's storage drive (like Hard drives, solid-state drives (SSD), or Non-Volatile Memory Express (NVMe)) as virtual memory. This extra space acts like temporary RAM, allowing more programs to run at the same time. Storage drives operate at lower data transfer speeds and have higher access times as compared to RAM; that's why the use of virtual memory can result in reduced system performance.

When several programs are open simultaneously, the operating system may transfer data from less active programs to virtual memory. This frees up space in RAM for the programs you are currently using, but may make those fewer active programs slower to respond.

DO YOU KNOW?

The first personal computer, the **IBM 5150** (1981), was equipped with only **16 KB** of RAM. In contrast, modern DDR5 RAM can transfer data at speeds exceeding **50 GB** per second, enabling the copying of a full high-definition movie in less than **one** second.

1.5 Processes vs Threads

A **process** is an independent program that is currently being executed by the computer. It has its own memory space CPU time, and other resources (such as files or network connections). Processes are isolated from one another to ensure stability and security; a problem in one process generally does not affect another.



A **thread**, on the other hand, is the smallest unit of execution within a process. Multiple threads can exist inside a single process, each performing a different task. All threads in the same process share the same memory and resources, but operate independently.

Example:

Consider a web browser:

- The entire browser application is one **process**.
- Within this process, different **threads** are responsible for:
    - One thread is loading and rendering a webpage.
    - The second thread is playing audio or video content.
    - Another thread is downloading files in the background.

By using the above multiple threads, the browser can continue loading new content while playing a video, without making the user wait for one task to finish before starting another.

1.5.1 Multithreading

Multithreading is an operating system technique that allows a single process to perform multiple tasks at the same time by dividing its work into smaller units called **threads**.

Each thread runs independently but shares the same memory and resources of the process, enabling faster execution, better responsiveness, and efficient use of system resources.

1.5.2 Benefits of Multithreading

Multithreading offers several advantages, and some of the most important are discussed below:

- **Enhanced Performance:** Tasks can be divided into multiple threads and executed in parallel, allowing complex operations to complete more quickly and improving overall system efficiency.
- **Improved Responsiveness:** Applications remain responsive even when one thread is busy with a specific task.
  Example: A word processor enables continuous typing while another thread checks spelling in the background.

- **Support for Concurrent (parallel) Operations:** Multiple tasks can progress during the same period, which is particularly important in applications such as games, video editing, and real-time communication tools.
- **Efficient Use of Resources:** Threads share the same memory space and resources of their parent process, requiring fewer system resources compared to creating separate processes.

1.6 System Calls

A **system call** is a request made by a program to the operating system to perform a specific task that the program cannot do directly. They act as a bridge between **user programs** and the **kernel**, allowing applications to access hardware and core OS functions safely. Without system calls, programs that directly control hardware can be unsafe and complex.

Example: When you save a file in a text editor, the program uses a system call to tell the operating system to write the data to the storage drive(like a Hard drive).

1.6.1 Types of System Calls

The main types of system calls include:

1. **open** Opens a file for reading or writing.
   Example: Opening a music file to play.
2. **read** Retrieves data from a file or input device.
   Example: Reading text from a document.
3. **write** Sends data to a file or output device.
   Example: Saving an image to the computer.
4. **fork** Creates a new process by duplicating an existing one.
   Example: Opening a new browser tab, where the OS may use fork to create another process.

DO YOU KNOW?

Linux and **macOS** have a few hundred system calls, while **Windows** uses nearly **2,000**. These calls handle everything from opening files to running apps and showing graphics in the background while you work.



1.7 File System Structure and Management

An operating system not only runs programs but also stores and organizes the users' data. This is achieved through the **file system**, which provides a structured way for users and applications to store, locate, and manage information on storage devices. A well-designed file system ensures that data remains organized, secure, and easily retrievable, even when there are thousands of files contain complex data.

1.7.1 Files

A **file** is a collection of related data stored on a computer. It may contain text, images, audio, video, or program instructions.

1.7.2 Folders

A **folder** (also called a directory) is a container used to organize files and other folders in a logical structure, making it easier to locate and manage information.

1.7.3 Metadata

Metadata refers to details about a file folder, such as its name, type, size, date of creation, and date of last modification. This information helps both the operating system and the user identify and manage files folders without opening them.

1.7.4 File Systems

A file system is the way an operating system stores and organizes files on a storage device, such as a hard drive, SSD, or USB. It decides where each file will be kept and save its location so it can be access later. Some widely used file systems include computer-based file systems are including:

- **FAT 32-** Used in USB flash drives
- **NTFS –** Used by Windows OS.
- **APFS / HFS+ –** Used by macOS.
- **EXT4 –** Commonly used by Linux OS.

The choice of file system affects performance, storage capacity, and security features.

Example: If you save a photo on your computer, the file system makes sure it is stored in the right place and knows exactly where to find it when you open it

again, just like a librarian placing a book in the right section and knowing where to locate it later.

ACTIVITY

Objective:
Demonstrate how system calls work when managing files.
Instructions:

1. **Prepare**:
   - Give each student a small card labeled with a file name (e.g., photo.jpg, homework.docx, song.mp3).
   - Write "**open**", "**read**", "**write**", and "**close**" on the board.
2. **Steps**:
   - Call one student the **"Operating System"**.
   - Another student plays the **"Application"** (e.g., Photo Viewer).
   - The Application requests: "OS, please **open** photo.jpg".
   - The OS responds by selecting the card from the "file list" and handing it over.
   - Repeat with **read**, **write**, and **close** for different files.
3. **Wrap-up Discussion**:
   - Explain that each action represents a **system call**.

1.8 Types of Operating Systems

Operating systems are designed according to the needs of the device and the work it performs. Each type serves a different purpose and is optimized for specific tasks.

1.8.1 Real-Time Operating System (RTOS)

A **Real-Time Operating System** is designed to process data and respond within a strict time limit, known as a **deadline**. It is used where even a tiny delay can cause system failure or serious consequences.

- **Key Feature:** Processes tasks immediately as they arrive, without long waiting times.
- **Used in:** Critical systems such as air traffic control, heart-monitoring devices, or industrial robots.

1.8.2 Embedded Operating System (EOS)

An **Embedded OS** is a small and highly efficient operating system built into a specific device to control only the functions it needs. It is not meant for general-purpose computing but is optimized for one task or a small set of tasks.

- **Key Feature:** Uses very little memory and power, and is often stored permanently inside the device.
- **Used in:** Home appliances (microwaves, washing machines), printers, smart TVs, and ATMs.

1.8.3 Network Operating System (NOS)

A **Network OS** manages and supports multiple computers connected through a network. It enables the sharing of resources like files, printers, and internet connections among users like windows multipoint servers.

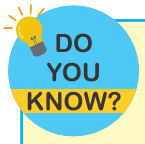
- **Key Feature:** Focuses on communication and coordination between computers.
- **Where Used:** Offices, schools, and data centers.

  Example: In a school computer lab, students use different computers, but all can save files to the same server and print from the same printer, thanks to the Network OS.

1.8.4 Mobile Operating System

A **Mobile OS** is designed for smartphones, tablets, and other handheld devices. It is optimized for **touch-screen use**, **battery saving**, and **mobile apps**.

- **Key Feature:** Supports wireless connectivity, cameras, sensors, and app stores.
- **Where Used:** Smartphones, tablets, smartwatches.



The very first mobile operating systems, like **Symbian** (used in early Nokia phones), could only run a few small apps. Today's mobile OS platforms like **Android** and **iOS** can handle millions of apps, 3D games, and even professional video editing tools.

EXERCISE

Multiple Choice Questions

1. **Which is NOT an example of an operating system?**
   (a) Linux (b) Windows
   (c) Photoshop (d) macOS
2. **In multi-user environments, the OS ensures:**
   (a) Equal access and data privacy (b) All users share the same files
   (c) Hardware is never used (d) Internet speed is increased
3. **In Windows, user accounts can be created via:**
   (a) Task Manager (b) Control Panel or Settings
   (c) Disk Management (d) BIOS settings
4. **The core part of the OS that interacts directly with hardware is the:**
   (a) Shell (b) Kernel
   (c) Device Driver (d) System Library
5. **A graphical shell allows the user to:**
   (a) Type commands only (b) Click icons and use menus
   (c) Interact only via code (d) Access only the BIOS
6. **In the FCFS scheduling method, processes are served:**
   (a) Randomly (b) Shortest job first
   (c) In the order they arrive (d) By priority only
7. **Threads in the same process:**
   (a) Have separate memory spaces (b) Share the same memory and resources
   (c) Run on different OS (d) Cannot run simultaneously

8. **Which system call is used to create a new process?**

(a) Open (b) read
(c) Write (d) fork

Short Questions

1. Why is the operating system referred to as the "central controller"?
2. How does the OS act as a translator between the user and hardware?
3. Define one way to create a new user account in Windows.
4. How is kernel different from shell.
5. Give one example of a system library and its purpose.
6. Explain one advantage and one disadvantage of FCFS scheduling.
7. Why is virtual memory slower than RAM?
8. Define a system call.
9. What is the role of a file system?
10. Define multitasking in computer system.

Long Questions

1. Describe the architecture of an operating system, explaining kernel, shell, and layered design with examples.
2. Explain the process lifecycle in detail with examples.
3. Differentiate between RAM and virtual memory, and explain how multithreading improves performance.
4. Describe system calls, their purpose, and give examples of at least three types.
5. A computer system uses the First-Come, First-Served (FCFS) scheduling method to manage processes. The following table shows the arrival time and the CPU burst time (time needed for execution) of three processes:

Process	Arrival Time (seconds)	Burst Time (seconds)
P1	0	4
P2	1	3
P3	2	1

Requirements:

I. Arrange the processes in the order they will be executed according to the FCFS scheduling method.
II. Show the execution sequence of all processes.
III. Calculate the start time and completion time for each process.
IV. Find the waiting time for each process.
V. Calculate the average waiting time for all processes.

Answer Key for Multiple Choice Questions

1. **C –** Photoshop
2. **A –** Equal access and data privacy
3. **B –** Control Panel or Settings
4. **B –** Kernel
5. **B –** Click icons and use menus
6. **C –** In the order they arrive
7. **B –** Share the same memory and resources
8. **D –** fork

UNIT 2

System Recovery and Advanced Maintenance

Student Learning Outcomes

By the end of this chapter, you will be able to:

- Define post-troubleshooting maintenance and explain its role in preventing recurring system issues.
- Use Task Manager, Resource Monitor, Resource Monitor and to monitor CPU, memory, and disk usage, and terminate unresponsive applications by reviewing system logs .
- Start Windows in Safe Mode and explain when this mode is useful.
- Create and use a restore point to return the system to a previous working state.
- Distinguish between System Restore and System Reset, and identify when to use each.
- Explain the purpose of bootable recovery media and when it is required.
- Create a bootable USB using Rufus or built-in Windows tools, choosing the correct partition style (GPT or MBR).
- Identify the differences between GPT and MBR and check a disk's partition style.
- Explain the role of BIOS and UEFI in system startup.
- Describe the Power-On Self-Test (POST) process and its importance.
- Access BIOS/UEFI settings to change boot order or reset defaults.
- Recover deleted files using built-in tools as well as third-party recovery tools such as Tenorshare.
- Perform Disk Cleanup to remove temporary and unnecessary files.
- Apply software updates (patch management) to improve security and stability.
- Run virus scans using Windows Defender and enable real-time protection.
- Create a maintenance calendar to schedule regular updates, cleanups, and scans.
- Maintain repair and update logs using a standard template.
- Explain the importance of keeping accurate system maintenance records.

Introduction

In this chapter, we will explain how to maintain and recover a computer system after troubleshooting. You will learn to monitor system health using tools like Task Manager, Resource Monitor, and Event Viewer, and explore recovery options such as Safe Mode, System Restore, and bootable drives.

The chapter also introduces the role of Basic Input/Output System (BIOS) and Unified Extensible Firmware Interface (UEFI) in system startup, simple data recovery techniques, and important preventive maintenance tasks like disk cleanup, updates, and virus scans. By the end of this chapter, you will understand how to keep system logs and maintenance records to run a computer system smoothly and handle common recovery tasks effectively.

2.1 Introduction to Post-Troubleshooting

Troubleshooting steps, including restarting the system, checking physical connections, and addressing simple issues related to software or hardware and to fix common computer problems, The next step is to learn what actions to take once the issue is resolved. This process is called *post-troubleshooting* maintenance. It includes additional steps that help the computer continue to work properly and avoid the same problems in the future. This chapter explains how to maintain a computer system and keep it stable, secure, efficient and reliable.

2.1.1 Importance of Continuous Maintenance

Solving a computer problem is only the first step. The system still needs regular care to continue working properly. Without maintenance, the same issue can happen again, or new problems may appear. Regular maintenance keeps the system stable and prevents future problems. It ensures better performance, longer life, and fewer disruptions in computer usage.

Example: If the computer becomes slow due to full storage, deleting a few files may improve its speed for a short time. However, if unnecessary files are not removed regularly or the system is not updated, the same issue may occur again.

2.1.2 Impact of Poor Maintenance

If a computer is not maintained regularly, it may face several problems that affect its performance and reliability. The following issues commonly occur due to a lack of proper maintenance:

Below are some common problems that result from a lack of regular maintenance:



- **Slow Performance:** Temporary files, unused programs, and background processes can build up over time. If these are not managed, the computer may become slow and unresponsive.
- **System Failures:** If important system updates are not installed, the computer may stop working properly. It can freeze, crash, or fail to start.
- **Security Risks:** When the system or apps are not updated, harmful software like viruses can enter the computer more easily.
- **Data Loss:** Without backups, important files can be lost during crashes or hardware failure.
- **Hardware Damage:** Dust and overheating can damage the hardware components if the system is not cleaned or properly ventilated.

2.1.3 Introduction to System Health Monitoring

For the smooth running of a system, it is important to check its performance regularly. This process is known as system health monitoring. Modern operating systems, such as Microsoft Windows, include built-in tools that help monitor system activity.

These tools allow you to:

- Check how much memory and processing power are being used
- Monitor background applications and services
- View system warnings and error messages
- Identify programs that may slow down the computer

Regular monitoring helps detect small issues before they become serious. By using these tools, users can take timely actions to improve performance and avoid future problems.

2.2 Using Built-in Diagnostic Tools

In addition to basic troubleshooting steps, modern operating systems provide built-in tools to help users monitor system performance in more detail. These tools give useful information about the computer's current state and help identify problems related to memory, storage, processing power, and system errors.

This section introduces three important diagnostic tools available in Microsoft Windows:

- Task Manager

- Resource Monitor
- Event Viewer

Each of these tools plays a different role in resolving system health and performance-related issues.

2.2.1 Task Manager

Task Manager is a built-in tool in Microsoft Windows that helps users to monitor the performance of their computers. It shows real-time information about how the system's processor (CPU), memory (RAM), and storage (disk) are being used. The Processes tab of the task manager shows all active programs and how much system resources they are using, while the *Performance* tab displays detailed graphs for CPU, memory, disk, and network usage.

How to Open Task Manager?

There are two simple ways to open Task Manager:

I. Press the **Ctrl + Shift + Esc** shortcut key from the keyboard
II. Or right-click on the taskbar and select Task Manager from the menu

What You Can Do with Task Manager?

Task Manager is helpful when your computer is slow, or an application stops responding. It allows you to:

- See how much processing power and memory are being used by each program.
- Select a program from the processes tab of task manager and click End Task to close it if it is not working properly (see **Figure 2.1**).
- View programs running silently in the background.



Figure 2.1: Task Manager Showing Active Processes and System Performance in Windows

- The performance tab of the task manager shows real-time graphs and data showing how your computer is performing (see **Figure 2.1**).

2.2.2 Resource Monitor

Resource Monitor is an advanced system tool in Microsoft Windows that gives a deeper view of how your computer's resources (memory, CPU, and disk) are being used, as shown in Figure 2.1 While Task Manager gives a quick overview, Resource Monitor provides more detailed information about which programs and background services are using system resources. It is especially helpful when the system is running slowly, overheating, or using a lot of memory without any clear reason.

How to Open Resource Monitor?

There are two common ways to open Resource Monitor:

- Open Task Manager, go to the Performance tab, and click on Open Resource Monitor at the bottom.
- Or press the shortcut key **Windows + R**, type the command ***resmon*** in the dialog box that appears, and press Enter (see Figure **2.2**).

Key Features of Resource Monitor

Some key features of the resource monitor are listed below:

- **CPU Tab:** Shows which processes and services are using the processor.
- **Memory Tab:** Displays how much memory is used, free, or reserved.

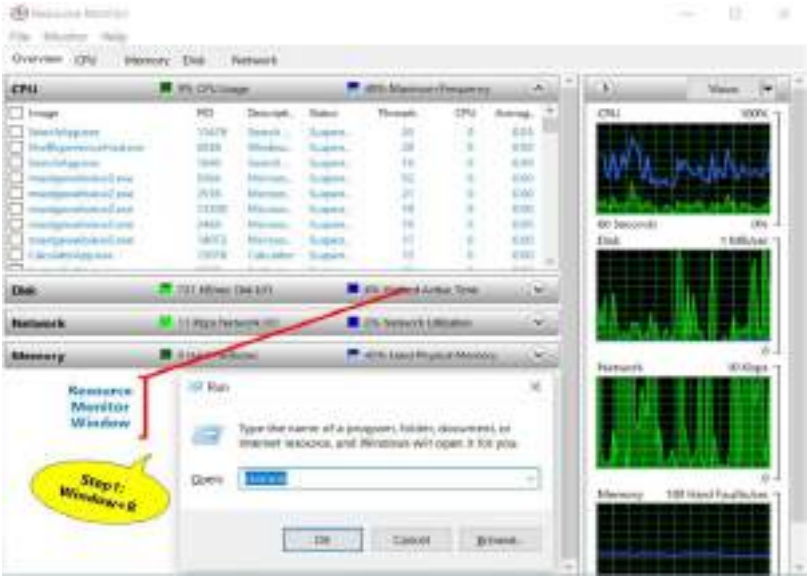


Figure 2.2: Resource Monitor to Monitor the Usage of System Resources

- **Disk Tab:** Monitors which programs are reading from or writing to the hard drive.
  Network Tab: Shows which applications are sending or receiving data over the internet or network.

2.2.3 Event Viewer

Event Viewer is also a built-in Windows tool that records detailed logs of system activities. It helps users check for errors, warnings, and important events related to software, hardware, and security, as shown in **Figure 2.3**.



Figure 2.3: The Event Viewer to View Upcoming Events Related to System Maintenance

How to Open Event Viewer

Press the shortcut key **Windows + R**, type the command ***eventvwr*** in the dialog box that appears, and press Enter.

ACTIVITY

Objective:

Use **Task Manager** to identify and end an unnecessary background program.

Steps:

1. Open **Task Manager** using **Ctrl + Shift + Esc**.
2. Go to the **Processes** tab.
3. Find a program running in the background that you are not using (e.g., Music app, update service).

4. Select it and click **End Task** (only if you're sure it is safe to close).
5. Observe if your computer's speed improves.

Answer the questions:

Did ending the task improve system performance?

Why do you think that happened?

2.3 System Recovery Options

In some cases, basic troubleshooting is not enough to fix a problem. The computer may crash, become unresponsive, or fail to start. To handle such situations, operating systems like Microsoft Windows provide special tools known as system recovery options. These recovery options help restore the computer to a working state without losing important data in most cases. In this section, we learn about the three commonly used recovery tools.

2.3.1 Safe Mode

Safe Mode starts Windows with only the most fundamental drivers and services. It disables unnecessary background programs and startup software, making it easier to fix problems caused by faulty apps or settings.

When and How to Use

Safe Mode is useful when:

- The system keeps crashing or restarting
- A program or driver is causing the computer to freeze
- You need to scan for malware or uninstall a recent app

How to open Safe Mode:

On most systems: Go to Settings > Update & Security > Recovery > Advanced Startup > Restart Now. Then choose Troubleshoot > Advanced Options > Startup Settings > Restart > Press **F4.**

On some older systems: Restart and press **F8** repeatedly before the Windows logo appears. Select safe mode, press enter.

Safe Mode with Networking

This version of Safe Mode includes network drivers, so you can access the internet. It is helpful when:

- You need to download updates or drivers
- You want to use online help tools
- You want to run a cloud-based antivirus scan

To enter it, follow the same steps as above and press **F5** instead of **F4**.

SMART TIPS

While using Safe Mode, follow these precautions:
Do not change settings unless you understand their purpose.

- Avoid deleting system files; they may be important for Windows to work properly.
- Use only built-in or trusted tools to fix issues.
- Restart your computer normally after finishing the task.
- Ask for help from your teacher or IT expert if unsure.

Safe Mode is powerful, but it should be used carefully.

2.3.2 System Restore

System Restore allows you to take your computer back to an earlier state without deleting personal files. It works by using restore points, which are saved versions of your system created before major changes.

What are The Restore Points?

A restore point is like a saved snapshot of your system's settings and files. It helps you return your computer to an earlier, working condition if something goes wrong, such as after installing a faulty program or driver. Restore points are made automatically before system updates or driver installations, or manually by the user when it is needed.

Steps to Create a Restore Point

Follow these steps to create a restore point manually:

- Click the Start button and open the ***Control Panel***.
- Go to ***System and Security***, then click ***System***.
- On the left, click ***System Protection***.
- In the new window, under the System Protection tab, click the Create button.
- Type a name for your restore point (e.g., "Before installing printer driver") and click Create.
- Wait a few seconds for confirmation. Your restore point is now saved.

Steps to Use a Restore Point

If your computer is not working properly, follow these steps to go back to a previous state:

- Open the ***Control Panel***.
- Go to ***Recovery*** and click ***Open System Restore***.
- A window will appear showing available restore points.
- Select the most recent restore point when your system was working fine.

- Click **Next**, then Finish.
- Your system will restart and restore the selected settings.

2.3.3 Bootable Recovery Media

Sometimes, a computer may stop working completely and fail to start. In such cases, normal recovery options like Safe Mode and Restore Points may not work. So, in this case, we use bootable recovery media to recover the system.

Understanding Bootable Media and Its Uses

Bootable recovery media is a ***USB*** flash drive or ***DVD*** that contains important system files. It helps start the computer from an external device instead of the built-in storage. After starting the system, you can use built-in recovery tools such as:

- Startup Repair to fix problems that prevent Windows from loading
- System Restore to go back to a working state using a restore point
- Command Prompt for advanced troubleshooting
- System Reset or Reinstallation to reinstall the operating system if needed

When We Use Bootable Media

We use this media in such cases where the system:

- Safe Mode is not accessible
- You need to repair or reinstall the operating system.

How to Create and Use a Bootable USB

To create a bootable USB:

- Download the system image or installer from the official website of your operating system provider (e.g., Windows ISO, Linux ISO, or macOS installer).
- Download Rufus (a free USB tool) from its official site (https://rufus.ie/en/), for Windows only. For macOS, you can use its built-in tools like Terminal or Disk Utility.
- Plug in a USB drive (16 GB or more).
- Open Rufus, select the USB and the ISO file.

For Windows or Linux systems, choose GUID Partition Table (**GPT)** for modern PCs and Master Boot Record (**MBR)** for older systems. Then click Start to create the bootable USB.

DO YOU KNOW?

GPT and MBR are two different ways of organizing data on a storage drive (Hard drive or SSD).

- MBR is the older system. It supports up to **2 TB** of disk size and a maximum of **4** primary partitions.
- GPT is the modern standard. It supports disks larger than 2 TB, allows over **100** partitions, and is required for UEFI-based systems.

That's why new computers and operating systems now prefer GPT over MBR!

How to Check the Partition Style (MBR or GPT)

To check the Partition Style, you need to:

i. Press Win + X and select Disk Management.
ii. Find the drive where Windows is installed (usually Disk 0).
iii. Right-click on the disk label (e.g., Disk 0), then choose Properties.
iv. Go to the Volumes tab.

Look at Partition Style:

It will show either:

- GUID Partition Table (GPT)

Master Boot Record (MBR)

SMART TIPS

If your computer uses GPT, but you create the USB in MBR format, it might not boot at all. The same happens if your system uses MBR and you use a GPT USB.

ACTIVITY

Objective:

Learn how to create a bootable recovery drive using built-in Windows tools.

Requirements:

- A Windows 10 or 11 PC
- An empty USB drive (at least **16** GB)

Steps:

I. Insert the USB into the computer.
II. Press the Windows key, type Recovery Drive, and open it.
III. Check "Backup system files to the recovery drive" and click Next.
IV. Select your USB drive and click Next (Do not click "Create" in class unless permitted).

Answer the question: What does Windows say this recovery drive will help you with?

2.4 BIOS/UEFI and Boot Process Awareness

When you press the power button on your computer, several hidden steps take place before the operating system (like Windows) appears on the screen. These early steps are controlled by special built-in software called ***firmware***. In this section, you will learn what firmware is, how the Basic Input/Output System (BIOS) and Unified Extensible Firmware Interface (UEFI) help start the computer, and how you can access basic startup settings if needed.

2.4.1 Role of BIOS / UEFI in Computer Startup

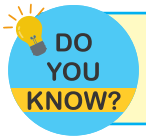
All computers have special software built into the motherboard called ***firmware***. It is not stored on the hard drive and does not get deleted when the computer is turned off. Firmware gives the computer basic instructions on how to start.

Two common types of firmware are:

- **BIOS** older firmware software shows a basic, text-only menu and works with the keyboard only.
- **UEFI** newer type of firmware software, supports a graphical menu that uses both mouse and keyboard, and larger hard drives.

Role of BIOS/UEFI in Starting the Computer:

- Checks if the hardware is working correctly (like RAM, keyboard, and hard drive).
- Finds the boot device (USB, hard disk, etc.) and starts the operating system.
- Allows users to change system settings like date, time, and boot order.



BIOS was first developed in the 1980s, while UEFI is now used in most modern computers because it is faster and more secure.

2.4.2 The POST Test

POST stands for Power-On Self-Test. It is the first thing the firmware does after the computer is turned on. POST checks the most important hardware to make sure everything is working properly. If all parts are fine, the computer continues to load the operating system. If something is wrong, it may beep or show an error message on the screen.

Example: If the RAM is missing, the computer will beep continuously.

Accessing BIOS and resetting to defaults

Sometimes you need to enter BIOS/UEFI to change settings or fix problems.

How to Open BIOS/UEFI Settings:

To open this setting, you need to shut down your computer.

i. Turn it on and immediately press a key like **Del**, **F2**, **F10**, or **Esc** (the correct key depends on the brand, like HP, DELL, and MSI).

ii. The BIOS or UEFI screen will appear.



Figure 2.4: BIOS /UEFI

Resetting to Default Settings

If the wrong settings are saved, you can restore the system to its original state by selecting "Load Setup Defaults" or "Reset to Default" inside BIOS/UEFI settings.

2.4.3 Boot order and boot repair

Boot order means the sequence in which the computer looks for a device to load the operating system. For example, it may try the USB drive first, then the hard drive.

How to Change Boot Order:

- Enter BIOS/UEFI settings.
- Go to the Boot tab or menu.
- Move your desired device (e.g., USB) to the top of the list.
- Save changes and exit.

Sometimes the computer cannot start due to missing files or errors. This can be fixed by using Startup Repair from Windows recovery USB or DVD (see Section **2.3.3**).

2.5 Data Recovery Techniques

Sometimes, we delete files by mistake or lose them due to system problems. In many cases, it is possible to recover lost files using built-in tools or with the help of special third-party software. This section explains basic methods for file recovery and how to avoid permanent data loss. There are two main ways to recover deleted files:

i. Using built-in tools.
ii. Using third-party software.

2.5.1 Built-in Tools

These tools are already part of the operating system, which is why we call them built-in tools. In this section, we will discuss some built-in tools that are available to recover deleted files.

Recycle Bin

When a file is deleted in Windows, it usually goes to the Recycle Bin. It stays there until the bin is empty. You can open the Recycle Bin, right-click the file, and select Restore to bring it back to its original location.

File History

File History is a backup feature in Windows. If it is turned on, the system keeps copies of your files at regular intervals. You can recover files from earlier versions, even if they have been deleted or changed. To turn it on, go to Settings > Update & Security > Backup, connect an external USB or Hard drive, and select "Add a drive". Once enabled, Windows will regularly back up your files like Documents, Pictures, and Desktop on that drive.

Previous Versions

This feature allows you to recover an earlier version of a file or folder. Right-click the file or folder, choose Properties, then go to the Previous Versions tab. If available, you can select and restore an older copy.

2.5.2 Third-party tools

Sometimes, deleted files are not found in the Recycle Bin or Previous Versions. In such cases, you can try using third-party recovery software. ***Tenorshare*** is a free and easy-to-use file recovery program demonstrated in **Figure 2.5**. It can scan your hard drive, USB, or memory card to find files that were recently deleted. You can then choose which files to recover.

How to use Tenorshare?

Steps to Use Tenorshare :

i. Open Tenorshare, select the location from which you want to recover a file, such as a specific drive or folder(see **Figure 2.5**).
ii. Choose the type of file to recover (e.g., pictures, documents).
iii. Click "Scan" to start searching for lost files.
iv. Select the files you want to recover and click "Recover" to restore them.

Recovery success depends on how recently the file was deleted and whether new data has overwritten it.



Figure 2.5: Tenorshare Data Recovery Tool

2.5.3 Limitations of recovery and best practices

While data recovery tools can be helpful, they have the following limitations.

- If a file is deleted a long time ago, it may no longer be recoverable.
- Recovery is not always possible if the hard drive is damaged or formatted.
- Files may be permanently lost if the Recycle Bin is emptied, and no backups were made.

SMART TIPS

1) Never install recovery software on the same drive where your lost file was stored. This may reduce the chance of successful recovery.
2) Always check the Recycle Bin first before trying advanced recovery tools. It is the easiest and safest way to restore deleted files.

ACTIVITY

Objective:
To help students understand how file recovery works by using Tenorshare on a test file.
Steps:
1) Create a text file named "TestFile.txt" on the Desktop.
2) Delete it and empty the Recycle Bin.
3) Open Tenorshare → Select All Files → Choose Desktop as location.
4) Click Scan → When the file appears, select and click Recover.
5) Save it to another folder (e.g., Documents).

Answer the questions:
Were you able to recover the file?
Why should you avoid saving new files to the same drive after deletion?

2.6 Best Practices for Preventive Maintenance

Just like we take care of our health or maintain our vehicles, computers also need regular attention to keep working properly. Preventive maintenance means doing simple tasks that help avoid bigger problems in the future. These tasks can improve your computer's speed, keep it secure, and help it last longer.

2.6.1 Disk Cleanup

After some time, the computer collects unnecessary files such as temporary internet files, system cache, Windows update files, and items in the Recycle Bin. These files take up space and may slow down system performance.
Disk Cleanup is a built-in Windows tool that helps remove such files safely.

Steps to Perform Disk Cleanup

I. Type Disk Cleanup in the Windows search bar and open the tool from the list. This opens a small window asking you to choose a drive to clean, as shown in **Figure 2.6**.
II. Select the drive you want to clean (usually C:) and click OK.
III. Check the boxes from the dialog box that appears after OK, as shown in **Figure 2.6**, for the types of files you want to delete.
IV. Click OK to remove the selected files.

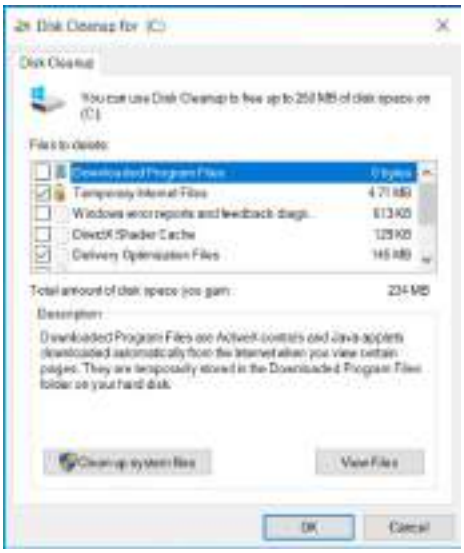


Figure 2.6: Disk Cleanup Menu

2.6.2 Patch management/ System Updation

Software updates, also known as patches, are released regularly by developers to fix bugs, add features, and improve security. Keeping your system updated helps to protect it from viruses and security threats.

There are two types of updates:

I. Operating system updates (for Windows, macOS, etc.)
II. Application updates (for programs like browsers, media players, etc.)

To check for updates in Windows:

- Open Settings > Update & Security.
- Click Check for updates.
- Install the updates if available.

2.6.3 Staying Protected with Windows Defender

Windows Defender is the built-in security tool in Windows that protects your computer from viruses, malware, and other threats. It offers two types of scans.

- Quick Scan to check common problem areas.
- Full Scan to check the entire system thoroughly.

Windows Defender also provides real-time protection, which means it keeps watching your system in the background and alerts you if anything suspicious happens.



2.6.4 Keeping a maintenance calendar

It's a good idea to set a regular schedule for basic maintenance tasks. A maintenance calendar helps you remember when to clean your system, scan for viruses, or check for updates. You can write it down or set reminders on your phone or computer.

2.7 System Documentation and Logs

Keeping proper records is an important part of a computer system's maintenance. When we document what we do, such as installing software, fixing issues, or performing updates, it becomes easier to manage and troubleshoot systems in the future. This section explains the basics of system documentation in a simple and practical way.

EXERCISE

Multiple Choice Questions

1. **Task Manager is mainly used to**
   (a) Monitor running applications (b) Clean up disks
   (c) Scan for viruses (d) Manage files
2. **Safe Mode starts Windows with**
   (a) All drivers and services (b) Basic drivers only
   (c) Network services (d) Antivirus tools
3. **Bootable USB drives are used for**
   (a) Installing games (b) System repair and recovery
   (c) Watching videos (d) Sending emails
4. **In UEFI systems, a bootable USB must be created in**
   (a) NTFS format (b) MBR style
   (c) GPT style (d) ZIP format
5. **The Power-On Self-Test (POST) is performed**
   (a) During system shutdown (b) After login
   (c) At system startup (d) When installing software

6. Tenorshare is a tool used to

(a) Play videos
(b) Recover lost files
(c) Create charts
(d) Monitor CPU

7. File History helps to

(a) Track website history
(b) Save copies of personal files
(c) Install apps
(d) Scan for malware

8. Windows Defender provides

(a) File compression
(b) Virus protection
(c) Disk formatting
(d) Music playback

Short Questions

1. Write one key function of the Resource Monitor.
2. Define the use of Safe Mode in system recovery.
3. List the steps to create a bootable USB using Rufus.
4. What happens during the Power-On Self-Test (POST).
5. Why is the boot order setting important when using recovery tools from a USB drive?
6. List the two benefits of running regular disk cleanup.
7. How does Windows Defender help in maintaining system health?

Long Questions

1. Explain the roles of Task Manager and Event Viewer in system maintenance. Give one example for each tool where it can help solve a problem.
2. Your computer is not starting properly due to a system issue. Describe how System Restore or a bootable recovery USB can be used to fix the problem without deleting personal files.
3. Briefly explain any two preventive maintenance techniques that help to keep a computer running smoothly.
4. Explain how deleted files can be recovered using a third-party software. Also, discuss why recovery may not always be successful.

5. A classmate wants to clean up disk space after a large Windows update. Explain how to use Disk Cleanup, including the option to remove system files safely.

Answer Key for Multiple Choice Questions

1. **A** – Monitor running applications
2. **B** – Basic drivers only
3. **B** – System repair and recovery
4. **C** – GPT style
5. **C** – At system startup
6. **B** – Recover lost files
7. **B** – Save copies of personal files
8. **B** – Virus Protection

UNIT 3

Introduction to Python Programming

Student Learning Outcomes

By the end of this chapter, you will be able to:

- Understand basic programming concepts and set up a Python development environment.
- Write and interpret basic Python syntax and structure, including variables, data types, and input/output operations.
- Use various operators and expressions in Python, including arithmetic, comparison assignment, logical operators and operator precedence.

Introduction

Welcome to the world of Python programming! Python is a popular language known for its simplicity and readability, making it ideal for both beginners and professionals. In this chapter, we will start with the basics, including setting up your development environment and learning fundamental programming concepts. By the end of this chapter, you will have a comprehensive understanding of Python, enabling you to write, test, and debug your own programs.

3.1 Introduction to Python Programming

Python is a widely-used high-level programming language famous for its simplicity and readability. It is versatile language and applicable to various fields, including web development, data analysis and artificial intelligence etc. Whether you are creating a simple script or developing a complex software, Python's user-friendly nature helps you get started quickly and efficiently.

3.1.1 Basic Programming Concepts

Computer programming is the process of creating a set of instructions that tell a computer how to perform a task. These instructions are written in a programming language that the computer can understand and execute. Think of computer programming like giving directions to a friend on how to reach your house. You need to be clear and precise so your friend doesn't get lost. Similarly, when we write programs, we give clear and precise instructions to the

computer to complete specific tasks.

Setting Up Python Development Environment

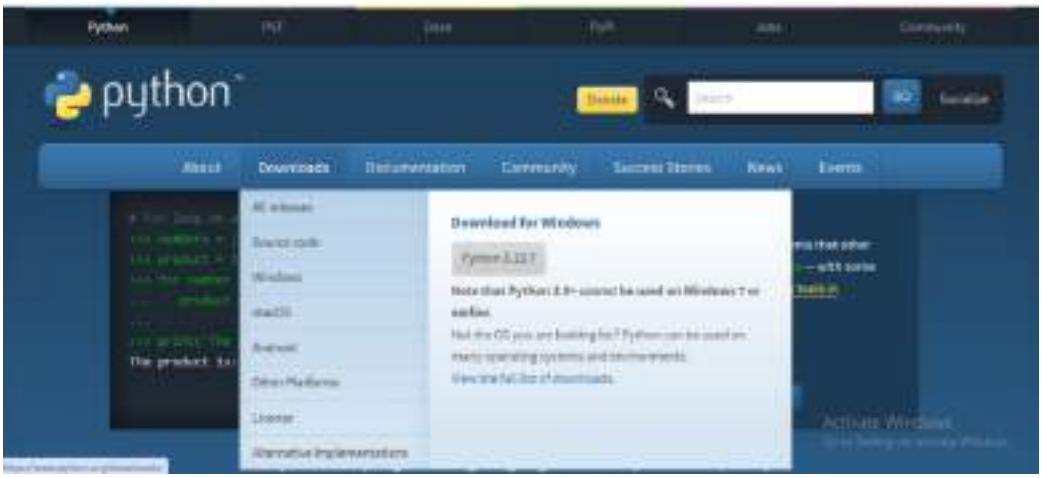
The development environment refers to the process of preparing a computer to write, run, and debug Python code effectively. This involves installing and configuring the necessary software, tools, and libraries that make development smoother and more efficient. We can download and install Python from https:

TOOL TIP

When installing Python, make sure to check the box that says "Add Python to PATH." This makes it easier to run Python from the command line. We can also use online services to write and run Python program. To complete the installation process of Python, follow these steps:

1. Download Python:

Visit https://www.python.org/. Navigate to the "Downloads" section and click on the latest version of Python (e.g., Python 3.13.x) for your operating system (Windows, macOS, or Linux).

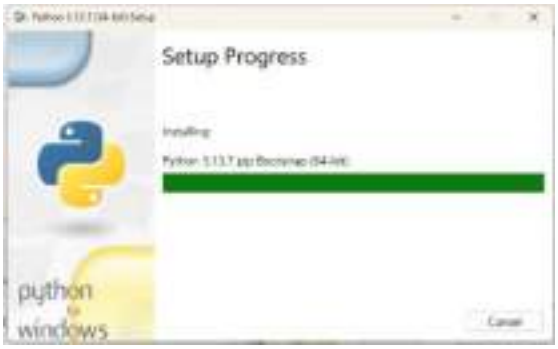


The installer will download automatically.

2. Install Python:

- Run the downloaded .exe file.

- Check the box "Add Python to PATH" during installation.
- Select "Install Now" or customize the installation if needed (e.g., choose the installation directory).
- Wait for the installation to complete, then click "Close".



3. Set Up an IDE:

- Download and install a popular IDE like PyCharm, VS Code, or IDLE (included with Python).
- PyCharm is a powerful IDE for Python development, available in a free Community edition or a paid Professional edition. Visit



https://www.jetbrains.com/pycharm/ to download the version suitable for your operating system (Windows, macOS, or Linux).

4. Configuring the IDE with Python

- Open PyCharm and start a new project.
- In the "Configure" or "Settings" menu, go to "Project Interpreter."
- Click the gear icon, select "Add Interpreter," and choose the path to your Python executable (e.g., C:\Python312\python.exe on Windows or /usr/bin/python3 on



Linux/macOS). Browse to your installation folder if it's not auto-detected.

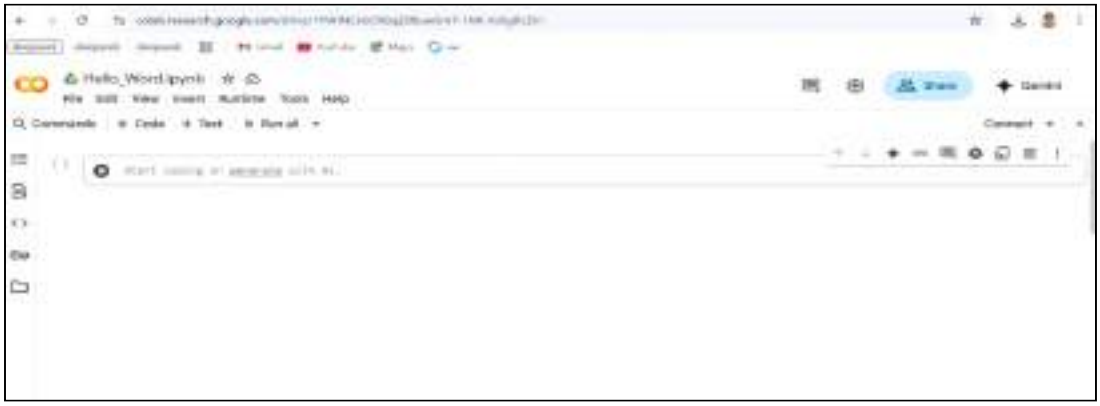
- Apply the settings and let PyCharm index the interpreter.

5. Testing the Configuration

- Create a new file (e.g., test.py) in your IDE.
- Write a simple script: print("Hello, Python!").
- Run the script (use the "Run" button in PyCharm/IDLE or F5 in VS Code after setting up a task).
- If the output appears (e.g., "Hello, Python!"), your IDE is correctly configured.

Google Colab

Another excellent option for using Python is Google Colab. It is a free cloud-based platform that provides a Jupyter notebook environment with pre-installed Python and popular libraries like NumPy, Pandas, and Matplotlib. To



get started, visit https://colab.research.google.com/, sign in with a Google account, and create a new notebook. No local installation is required, making it ideal for beginners or those working on machines with limited resources. You can write and execute Python code in cells.

3.2 Basic Python Syntax and Structure

The following Python program demonstrates the simplicity and readability of the language:



In this example, the print function is utilized to output the message enclosed in double quotation marks. This illustrates Python's straightforward syntax, where functions like print are used to perform actions such as displaying text.

Python Comments

Comments are the lines that are not executed by the Python interpreter. They are used to provide explanations or notes for the Programmer. Single-line comments start with the # symbol.

```
     # This is a single - line comment
     print ( "K2 is the second-highest mountain in the
     world " )
     '''
     This is a comment.
     '''
     print ( "Edhi Foundation operates the world's largest
    volunteer ambulance network." )
'''
Output:
K2 is the second-highest mountain in the world
Edhi Foundation operates the world's largest volunteer
ambulance network.
'''
```

Interesting Information

We can use ''' ...''' (Triple Quotes) in python for multiline comments but actually they are multiline string literals which are ignored at runtime.

3.2.1 Variables in Python

A variable in programming functions as a storage container within a computer's memory. It allows the storage of data that can be retrieved later in the code. In the example below, the variable age is utilized to hold numerical data. The value of a variable can change throughout the execution of a program, which is why it is referred to as a "variable":

```
     print ( age ="Hello , dear students ! ")
     age = 71
     print( "Azam lived for", age, "years")
     age = 60
     print ( " Iqbal lived for", age, "years")
```

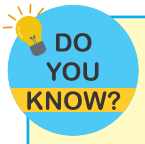
```
# Output
# Azam lived for 71 years
# Iqbal lived for 60 years
```

This example illustrates how the variable "age" is assigned different values to represent the ages of Azam and Iqbal, which are then printed as part of the output.

Variable Naming Rules in Python

Variable names in Python must adhere to the following rules:

- The name must begin with a letter (a-z, A-Z) or an underscore (_).
- Subsequent characters can include letters, digits (0-9), or underscores ( ).
- Variable names are case-sensitive, meaning age and Age are considered two different variables.
- Python's reserved keywords, such as for, while, if, etc., cannot be used as variable names.



Following are some of the reserved words in Python:

- if, else, elif, for, while, break, continue, and, or, not, in, is, True, False, None.
- We Can not use them as a variable name

Types of Variables

In Python, you can create variables of different types to store various kinds of data. Here are some common types of variables:

- **Integer (int):** Stores whole numbers. Example: age = 17
- **Floating-point (float):** Stores decimal numbers. Example: price = 19.99
- **String (str):** Stores text. Example: name = "Ali"
- **Boolean (bool):** Stores True or False. Example: is_student = True

Input and Output Operations in Python

Input and output operations allow you to interact with the user. You can ask the user to enter data (input) and display information to the user (output).

Input: We use the **input ()** function to get user input. The input () function displays a message on the screen and waits for the user to type something and press Enter. The text entered by the user is then stored in a variable. For example:

```
name = input("Enter your name: ")
print (" Hello, " + name + "!")
# Output
# Enter your name: Maryam Ashraf
# Hello, Maryam Ashraf!
```

This line of code asks the user to enter their name and stores it in the variable name.

Output: Use the print () function to display information on the screen. The print () function takes one or more arguments and displays them. For example:

This line of code displays a greeting message that includes the user's name.

Handling Integer and Float Inputs

To handle numeric inputs, you use the int() or float() functions to convert input strings to integers or floating-point numbers.

> In python indentation defines structure scope your code. Incorrect indentation may cause in your code.

Integer Inputs

```
# Example : Handling integer input
  user_age = int(input("Enter your age: "))
  print("Your age is:",user_age)
# Output
# Enter your age: 16
```

Float Inputs

```
# Example: Handling float input
user_height = float(input("Enter your height in meters:
") )
print("Your height is", user_height,"meters”)

# Output
# Enter your height in meters: 1.5
# Your height is 1.5 meters
```

3.3 Operators and Expressions

Operators are symbols that perform operations on operands. An expression is a combination of operators, and values that produces a result. Let's explore



different types of operators in Python.

3.3.1 Arithmetic Operators

Arithmetic operators are used to perform basic mathematical operations such as addition, subtraction, multiplication, division, modulus, exponentiation, and floor division as shown in the following code.

```
# Define variables
a= 10
b= 3
# Perform all arithmetic operations on these numeric
variables and print results
print(a, "+", b, "=", a + b) # Output: 10+3=13
print(a, "*", b, " = " , a * b) # Output: 10 * 3 = 30
print(a, "/", b, " = ", a / b)
# Output: 10 / 3 = 3.3333333333333335
print(a, "//", b, " = ", a // b) #floor division
# Output: 10 // 3=3
print(a, "%", b, " = ", a % b)
# Output: 10 % 3 = 1
print(a, "**", b, " = ", a ** b) # ** represent power
                                    operator:
# Output: 10**3= 1000
```

3.3.2 Comparison Operators

Comparison operators are used to compare two values or expressions. They determine the relational logic between them, such as equality, inequality, greater than, less than, and so on. These operators return a Boolean value (True or False) based on the comparison result. Here's a Python program that demonstrates the usage of all comparison operators.

```
Define variables
X = 10
Y = 5
# Greater than
print (x, ">", y, "=", x > y) # Output: 10 > 5 = True
# Less than
print (x, ">", y, "=", x < y) # Output: 10 < 5 = False
```





```
#Equal to
print (x, "==", y, "=", x == y) # Output: 10 == 5 = False
# Not equal to
print (x, "!=", y, "=", x != y) # Output: 10 != 5 = True
# Greater than or equal to
print (x, ">=", y, "=", x >= y) # Output: 10 >= 5 = True
# Less than or equal to
print (x, "<=", y, "=", x <= y) # Output: 10 <= 5 = False
```

The above code demonstrates Python's comparison operators by defining two variables, **x** and **y,** and comparing them using various operators like >, <, ==, !=, >=, and <=.

3.3.3 Assignment Operators

Assignment operators are used to assign values to variables. The most common assignment operator is the equal sign (=), which assigns the value on the right to the variable on the left. There are also compound assignment operators like +=, -=, *=, and /=, which combine arithmetic operations with assignment.

```
# Define initial values
a = 10
b = 5

# Assignment
x = a; print("a =" , x)# Output: a = 10
# Addition assignment
a += b; print("a after addition =" , a)# Output: a = 15
# Subtraction assignment
a -= b; print("a after subtraction =" , a)# Output: a = 10
# Multiplication assignment
a *= b; print("a after multiplication =" , a)# Output: a = 50
# Division assignment
a /= b; print("a after division =" , a)# Output: a = 10.0
#Floor division assignment
a //= b; print("a after floor division  =" , a)# Output: a = 2.0
# Modulus assignment
a %= b; print("a after modulus = " , a)# Output: a = 2.0
# Exponentiation assignment
a **= b; print("a after exponentiation =" , a)# Output: a = 32.0
```





Above code illustrates the use of compound assignment operators in Python. It starts by assigning a value to a variable "a". It then demonstrates various assignment operations including addition, subtraction, multiplication, division, floor division, modulus, and exponentiation. Each operation updates the value of "a" and the results are printed with the updated values. The code comments show the output of each operation.

3.3.4 Logical Operators

Logical operators are used to combine multiple or expressions. The most common logical operators are "and, or, and not", they are used to perform logical operations and return Boolean values based on the evaluation of the expressions involved.

```
# Define variables
x = True
y = False
# Logical AND
logical_and = x and  y
 print(x , "and ", y ,"=", logical_and) # Output: True and
False =False
# Logical OR
logical_or = x or y
print(x , "or " , y , " =", logical_or) # Output: True o
False = True
# Logical NOT
logical_not_x = not x
print("not", x, " = " , logical_not_x) # Output not True=
False
logical_not_y = not y
print("not", y, "=" , logical_not_y) # Output: not False=
True
```

The above code demonstrates the use of logical operators in Python. It defines two Boolean variables, x and y, and performs logical operations such as "and, or, not" on them. The results of these logical operations are printed, showing whether the expressions evaluate to True or False. The code also includes comments with the output of each logical operation.





3.3.5 Operator Precedence in Python

Operator precedence determines the order in which operations are performed in an expression. In Python as well as in Mathematics, certain operators have higher precedence and are evaluated before others. Understanding this helps ensure that your calculations are done correctly.

- **Parentheses '()':** Highest precedence. Operations inside parentheses are performed first. (3 + 2) * 4 evaluates to 20.
- **Exponentiation:** Performs power operations next.
  2 + 2 * 2 ** 3 = 32
- **Multiplication '*', Division** '/', **and Modulus '%':** These operations come next.
- **Addition '+' and Subtraction '-':** These have lower precedence compared to multiplication and division.
  For example: 4 * 3 + 10 / 2 – 11 % 3
  4 * 3 evaluates to 12, 10 / 2 evaluates to 5.0 and 11 % 3 evaluates to 2

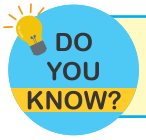
Perform add and subtract operators 12 + 5.0 – 2 = 15.

Example: Consider the following code to show operator precedence:

```
print("===  PYTHON  OPERATOR  PRECEDENCE  DEMONSTRATION
===\n")
# 1. Arithmetic Operators
print("1. ARITHMETIC OPERATORS:")
print(f"3 + 2 * 5    = {3 + 2 * 5}")   # Multiplication
before addition
print(f"(3 + 2) * 5 = {(3 + 2) * 5}")# Parentheses change
order
print(f"2 ** 3 * 4   = {2 ** 3 * 4}") # Exponentiation
before multiplication
print(f"15 / 3 * 2   = {15 / 3 * 2}") # Division and
multiplication (left to right)
print(f"15 // 4 + 2 = {15 // 4 + 2}")# Floor division
before addition
print(f"11 // 4 + 2 = {11 // 4 + 2}")# Floor division
before addition
print(f"17 % 5 * 2   = {17 % 5 * 2}") # Modulo before
multiplication
# Output
# === PYTHON OPERATOR PRECEDENCE DEMONSTRATION ===
```



```
# 1. ARITHMETIC OPERATORS:
# 3 + 2 * 5      = 13
# (3 + 2) * 5    = 25
# 10 - 3 + 2     = 9
# 2 ** 3 * 4     = 32
# 15 / 3 * 2     = 10.0
# 15 // 4 + 2    = 5
# 11 // 4 + 2    = 4
# 17 % 5 * 2     = 4
```



In python "f" before string is used for string interpolation (embedding variables to expression) inside a string.



TIDBIT

Exponentiation is right associative, meaning that when there are multiple ** operators, python evaluates then from rights to left.

Summary

- Python is a high-level, versatile programming language. It is known for simplicity and readability.
- Python is used in web development, data analysis and Artificial Intelligence etc.
- Programming involves giving precise instructions to computers. Programs tell computers how to perform specific tasks.
- Download Python from **python.org** (latest version)
- Installation steps: Download → Run installer → Check "Add Python to PATH" → Install
- IDE Options: PyCharm, VS Code, IDLE.
- Online Option: Google Colab (cloud-based, no installation)
- Arithmetic: +, -, *, /, //, %, **
- Comparison: >, <, ==, !=, >=, <=
- Assignment: =, +=, -=, *=, /=, //=, %=
- Logical: and, or, not
- Comments are used to improve code readability: #

EXERCISE

Multiple Choice Questions

1. **Which of the following steps is NOT part of the basic programming process?**
   (a) Write Code (b) Compile/Interpret
   (c) Execute (d) Ignore Errors
2. **What should you do when installing Python to run it from the command line more easily?**
   (a) Uncheck "Add Python to PATH"
   (b) Choose a different IDE
   (c) Check "Add Python to PATH"
   (d) Install only the IDE
3. **What will be the output of: print(10 // 3)?**
   (a) 3.333 (b) 3
   (c) 1 (d) 4
4. **What does the input() function return?**
   (a) Integer (b) Float
   (c) String (d) Boolean
5. **Which operator has the highest precedence in Python?**
   (a) Addition (+) (b) Multiplication (*)
   (c) Exponentiation (**) (d) Parentheses ()
6. **What is the result of 5 > 3 and 2 < 1?**
   (a) True (b) False
   (c) Error (d) None
7. **What does the += operator do?**
   (a) Adds two numbers (b) Compares values
   (c) Adds and assigns (d) Multiplies values
8. **Which function converts string integer?**
   (a) str() (b) int()
   (c) float() (d) input()

9. What is the output of the below code?

age = 25; print (" Age : " , age)

(a) Age: 25 (b) 25

(c) Age (d) age

10. Which symbol is used for comments in Python?

(a) // (b) /*

(c) # (d) --

Short Answer Questions

1. Why are comments important in programming?
2. Discuss three basic data types in python.
3. Describe the difference between integer and float data types in Python.
4. How does the if statement differ from the if-else statement?
5. What are logical operators and name all three?
6. How do you get user input in Python?
7. What is the purpose of print() statement in Python?
8. What is operator precedence? Give an example.
9. Write any three main rules for naming variables in Python?

Long Answer Questions

1. What are the basic data types in Python? Explain integers, floats, strings, and Booleans with examples.
2. What are arithmetic operators in Python. Provide examples of each operator and explain when you would use them.
3. Explain the input() and print() functions with examples.
4. Explain how comparison and logical operators work together in conditional statements. Provide a practical example.
5. Write a program in Python that will take a name from user and display greeting message to the given name.
6. Write a program in python that takes two numbers from user and display their sum.
7. Write a program in python that five numbers from user and display their average.
8. Solve the following expressions:
   a) 8 + 3 * 4 - 2 ** 2
   b) 20 % 7 * 3 + 10 // 3
   c) 2 * 3 ** 2 % 4 + 10 - 6 / 2

UNIT 4

Control Structures in Python

Student Learning Outcomes

By the end of this chapter, you will be able to:

- Implement control structures such as decision-making statements and loops in Python.
- Work with Python modules, functions, and built-in data structures like lists.
- Apply modular programming techniques and object-oriented programming concepts in Python.
- Handle exceptions, perform file operations, and apply testing and debugging techniques in Python.

Introduction

In programming, we often need to control the flow of our program based on different conditions or repeat certain actions multiple times. This is where control structures come into play. Control structures are essential because they help us manage the decision-making process and the repetition of tasks in our programs. There are two main types of control structures, Decision making and Looping:

4.1 Decision Making

Decision making in programming allows the program to choose different actions based on conditions. This is similar to how we make decisions in real life. For example, deciding whether to take an umbrella based on the weather situation. Python provide variety of conditional statements to implement decision making.

4.1.1 if Statement

The **if** statement lets us make decisions based on conditions. If the condition is true, it runs a block of code. In Python indentation defines the structure block or scope of your code. In Python proper indentation is mandatory. Incorrect indentation may cause error in your code (Indentation Error).

```
# Syntax of if statement if condition:
If condition:
        code to run if the condition is true
```



Example: If the temperature is above 30 degrees, we print a message.

```
temperature = 35                                    # Output:  It's a hot day
if temperature > 30:
    print("It's a hot day")
```

4.1.2 if-else Statement

The **if-else** statement allows us to execute one block of code if a condition is true and another block if the condition is false.

Syntax of if-else statement if condition:

```
If condition:
        code to run if the condition is true else:
else:
        code to run if the condition is false
```

Example:

```
temperature = 15                #Output: It's not a hot day
if temperature > 30:
   print("It's a hot day ")
else :
   print("It's not a hot day " )
```

4.1.3 Nested Conditions

Sometimes, we need to check multiple conditions inside another condition. This is called **nesting**.

Syntax of nested if statement

```
if condition1:
        if condition2:
            code to run if both condition1 and condition2 are true
else:
            code to run if conditionl is true but condition2 is false
else:
            code to run if conditionl is false
```

Example: If the weather is rainy and the temperature is below 15 degrees, we wear a raincoat. If it is only rainy, we take an umbrella. If the weather is not rainy, we just enjoy the day.





```
weather = "rainy"
temperature = 10
if weather == "rainy":
        if temperature < 15:
                print("Wear a raincoat")
         else :
                 print("Take an umbrella")
else:
          print("Enjoy your day!")
# Output: Wear a raincoat
```

Explanation: In this example, the code checks if the weather is rainy. If it is, it then checks if the temperature is below 15 degrees. If both conditions are true, it prints "Wear a raincoat". If the weather is rainy but the temperature is not below 15 degrees, it prints "Take an umbrella". If the weather is not rainy, it prints "Enjoy your day!".

4.2 Looping Constructs

Loops help us repeat actions, making our code more efficient and easier to read. There are two main types of loops in Python: **while** loops and **for** loops.

4.2.1 while Loop

A **while** loop runs as long as a condition is true. It checks the condition before each iteration and stops running when the condition is no longer true.

```
# Syntax of while loop while condition:
        code to run while the condition is true
```

Example: Add 1 to a number until it reaches 10.

```
number = 1
while number < 10:
    print(number)
    number += 1
```

```
'''
Output:
1
2
3
4
5
6
7
8
9
'''
```





Explanation: In this example, the code starts with a **number** set to 1. The 'while' loop checks if the **number** is less than 10. If it is, the program prints the current value of the **number** and then adds 1 to it. This loop continues until the **number** reaches 10, at which point the loop stops running.

4.2.2 for Loop

A **for** loop repeats a block of code a specific number of times. It is commonly used to iterate over a sequence (like a list, tuple, or string).

```
# Syntax of for loop
for variable in sequence:
code to run for each element in the sequence
```

Example: Say "As-Salaam-Alaikum" to each friend in a list of friends.

```
friends = ["Sami", "Raza", "Moosa"]
for friend in friends:
        print("Welcome to ", friend)
```

```
'''
Output:
Welcome to: Sami
Welcome to: Raza
Welcome to: Moosa
'''
```

Explanation: In this example, the code goes through each friend in the list and prints a greeting message for each one.

4.2.3 range() Function

We can use the **range()** function to generate a sequence of numbers, which is often used in **for** loops.

```
# Syntax of range function
range(stop)
range(start, stop)
range(start, stop, step)
```

Example: Print First 5 Whole Numbers.

```
for i in range (5):
     print (i)
```

```
'''
Output:
0
1
2
3
4
'''
```



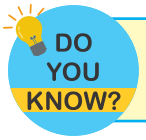
End statement

In Python, the end statement is not a separate command, but rather an optional parameter of the print() function. By default, every print() in Python ends with a newline character, but you can change what gets printed at the end using the end parameter.

Example 1: Print two Strings on same line using "end" statement.

```
print("Azhar Shah", end=" ")
print("Ahsan")
# Output:  Azhar Shah Ahsan
```

```
print("Ayyan", end="***")
print("Naveed")
# Output: Ayyan***Naveed
```



Strings in Python are surrounded by either Single quotation' ' marks or double quotation " " marks. Like 'Ayyan' is the same as "Ayyan".

Example 2: Print Numbers from 1 to 5 on the same line using "end" statement.

```
for i in range(1, 6):
  # if you want to print Numbers on same line use following line
          print(i, end=" ")
  # Output: 1 2 3 4 5
```

Explanation: The above code generates numbers from 0 to 4 and prints each number.



1. Write a **for** loop using **range()** to print the even numbers from 2 to 10 on single line.
2. Write a Python program that prints the first 10 multiples of 3 using a **for** loop and the **range()** function.

4.2.4 Nested Loops

Nested loops are loops inside other loop. For each iteration of the outer loop, the inner loop runs completely. We use nested loops when we need to perform repetitive tasks within tasks.

Example1: Nested loops for basic Number Pattern

```
print("=== Basic Number Pattern ===")
for i in range(3):       # Outer loop
    for j in range(4):   # Inner loop
        print(f"i={i}, j={j}")
    print("---")         # Separator after each outer loop
iteration
'''
# Output
=== Basic Number Pattern ===
i=0, j=0
i=0, j=1
i=0, j=2
i=0, j=3
---
i=1, j=0
i=1, j=1
i=1, j=2
i=1, j=3
---
i=2, j=0
i=2, j=1
i=2, j=2
i=2, j=3
--- '''
```

Example2: Nested loops to print stars (asterisk)in Triangular Pattern

```
print("=== Right-Angled Triangle ===")
n = 5
for i in range(1, n+1):
    for j in range(i):       # Inner loop runs 'i' times
        print("*", end=" ")
    print()  # New line after each row
'''
# Output
*
* *
* * *
* * * *
* * * * *
'''
```

4.3 Libraries in Python

Python offers an extensive standard library that includes built-in modules and data structures. A data structure refers to a particular format or method for organizing and storing data. In this section, we will examine the utilization of different libraries in Python.

4.3.1 Using Libraries in Python

In Python, libraries are like toolboxes full of useful tools that help you solve different problems without having to build everything from scratch. Libraries are like pre-built toolkits that you can use without having to write all the code yourself. Let's explore how to import and use different libraries with some simple examples.

Example: Import the random library to generate random numbers.

```
import random
# Generate a random number between 1 to 10
number = random.randint(1, 10)
print("The random number is:", number)
# Output:
# The random number is: 3
```

Example: Import the statistics library to perform statistical calculations.

```
 import statistics
# Calculate the mean of a list of numbers
data = [23, 45, 67, 89, 12, 44, 56]
mean_value = statistics.mean(data)
print("The mean value is:", mean_value)
# Output:
# The mean value is: 48
```

Explanation: The statistics library is a great tool for performing basic statistical calculations, such as finding the mean, median, and mode of a set of data. This can be particularly useful in data analysis tasks.

In short by using these libraries, you can save time and effort, allowing you to focus on solving the specific problem at hand rather than reinventing the wheel.



When importing libraries, make sure to only import what you need. You can explore the more libraries at https: //docs .python, org/3/

4.4 Lists in Python

In Python, a list is a versatile data structure that can hold a collection of items. You can create, access, and modify lists easily.

4.4.1 Creating Lists

A list is created by placing items inside square brackets [ ], separated by commas. Lists can contain items of different types, such as numbers, strings, or even other lists.

Example: Create a list of your favorite fruits.

```
fruits = ["Mango", "Apple" "Banana"]
 print(fruits)
# Output: ['Mango' 'Apple ' 'Banana ' ]
```

Explanation: This code creates a list named 'fruits containing three elements and then prints the list.

4.4.2 Accessing List Items

You can access items in a list by referring to their index, starting from 0.

Example: Access and print the second item from the list of fruits.

```
fruits = ["Mango", "Apple", "" Banana"
print(fruits [1])
# Output: Apple
```

Explanation: The code initializes a list 'fruits' containing 'Mango', 'Apple', and 'Banana', then prints the second item, 'Apple', using the index ' 1 '.

ACTIVITY

Write a Python program that imports the random and statistics libraries. Use random to generate a list of 10 random numbers between 1 and 50. Then, use statistics to calculate and print the mean (average) of those numbers.

4.4.3 Modifying a List

You can modify list items by accessing them via their index and assigning a new value.

Example: Change the first item in the list to "Orange" and add a new fruit "Pineapple".

```
fruits = ["Mango", "Apple", "Banana"]
fruits [0] = "Orange"
fruits.append("Pineapple")
print(fruits)
# Output: ['Orange', 'Apple', 'Banana', 'Pineapple']
```

Explanation: The code modifies the first element of the 'fruits' list to 'Orange', appends 'Pineapple' at the end, and prints the updated list.

4.4.4 Operations on Lists

Python provides several built-in methods to work with lists. Here are a few useful ones:

- append (item) - Adds an item to the end of the list.
- remove (item) - Removes the first occurrence of an item from the list.
- sort () - Sorts the list in ascending order.
- reverse () - Reverses the order of the list.

Example 1: Add a new student to the list of students and print the list.

```
students = ["Ahmed", "Sara", "Ali"]
students.append("Hina")
 print(students)
#Output : ['Ahmed', 'Ali', 'Sara', 'Hina']
```

Explanation: The code creates a list of students, adds 'Hina' to it, sorts the list alphabetically.

Example 2: Remove a student from the list of students and print the list.

```
# Remove a student from the list of students
students = ["Ahmed", "Sara", "Ali", "Hina", "Sara"]
students.remove("Sara")
print(students)
# Output: ['Ahmed', 'Ali', 'Hina', 'Sara']
```

Explanation: The code creates a list of students, removes the first occurrence of "Sara" from the list, and prints the updated list. Note that only the first matching item is removed.

Example 3: Sort a list of numbers in ascending order.

```
# Sort a list of numbers in ascending order
numbers = [34, 12, 89, 5, 23]
numbers.sort()
print(numbers)
# Output: [5, 12, 23, 34, 89]
```

Explanation: The code creates a list of numbers, sorts them in ascending order using the sort() method, and prints the sorted list.

Example 4: Reverse the order of fruits in a list.

```
# Reverse the order of fruits in a list
fruits = ["Apple", "Banana", "Orange", "Mango"]
fruits.reverse()
print(fruits)
# Output: ['Mango', 'Orange', 'Banana', 'Apple']
```

Explanation: The code creates a list of fruits, reverses the order of elements using the reverse() method, and prints the reversed list.

ACTIVITY

Imagine you are maintaining a list of your favorite books: ["To Kill a Mockingbird", "1984", "The Great Gatsby", "Pride and Prejudice"]. Perform the following tasks using Python:

- Add a new book "Alchemist" to the list.
- Replace "1984" with "Brave New World".
- Remove "The Great Gatsby" from the list.
- Merge this list with another list of books: ["War and Peace", "Hamlet"].
- Print the final list of books.

Write the Python code to execute these tasks and print the final list of books.

4.5 Testing and Debugging in Python

In Python programming, testing and debugging are essential practices to ensure that your code works correctly and efficiently.

4.5.1 Testing

Testing is the process of running your code with various inputs to check if it behaves as expected. The goal is to find and fix any issues before the code is used in real-world applications.

Types of Testing

- **Unit Testing:** Tests individual parts of the code (like functions or classes) in isolation. Python's unit test module is commonly used for this.
- **Integration Testing:** Checks how different parts of the code work together.
- **Functional Testing:** Validates that the software behaves as expected from the user's perspective.
- **Regression Testing:** Ensures that new changes don't affect the existing functionality.



Example: In this example, we use the unit test module to test the add function. This ensures that the function returns the correct results by checking various input values.

4.5.2 Debugging

Debugging is the process of finding and fixing errors (bugs) in your code. It involves identifying the root cause of problems and making the necessary changes.

Common Debugging Techniques

- **Print Statements:** Adding print statements to check the values of variables at different stages of the code.
- **Debugging Tools:** Using tools like pdb (Python Debugger) to step through the code, inspect variables, and understand the flow of execution.
- **Error Messages:** Reading and interpreting error messages to locate the source of the problem.



EXERCISE

Multiple Choice Questions

1. What does the range () function do in Python?

(a) Generates a list of numbers
(b) Creates a sequence of numbers
(c) Calculates the sum of numbers
(d) Prints a range of numbers

2. What is the output of the below code?

```
count = 0
for i in range(2):
for j in range(3):
 count += 1
print(count)
```

(a) 4
(b) 5
(c) 6
(d) 7

3. Which method is used to add an item at the end of the list in Python?

(a) insert()
(b) append()
(c) add()
(d) extend()

4. What is the purpose of control structures in programming?

(a) Only to make code longer
(b) To manage decision-making and repetition of tasks
(c) To create variables
(d) To import libraries

5. Which statement is used to execute code when a condition is false?

(a) if
(b) elif
(c) else
(d) while

6. How do you prevent print() from adding a newline?

(a) Use `end=""`
(b) Use `newline=False`
(c) Use `break`
(d) Use `continue`

7. Which library is used to generate random numbers?

(a) Datetime (b) statistics

(c) random (d) math

8. What does list.append() do?

(a) Removes the first item (b) Adds an item to the end

(c) Sorts the list (d) Reverses the list

9. What is the purpose of the range() function?

(a) To create a list of numbers

(b) To generate a sequence of numbers

(c) To calculate the range of values

(d) To sort numbers

10. Which debugging technique involves adding output statements?

(a) Unit testing (b) Print statements

(c) Error messages (d) Integration testing

Short Questions

1. Explain the use of the range() function in for loop?
2. What is the difference between the append and remove methods in Python lists?
3. Define debugging.
4. What are the three parameters of the range() function?
5. Name three list methods and their purposes.
6. How do you import a library in Python?
7. What is unit testing and why is it important?
8. How do you access the third element in a list?
9. What is the purpose of the end parameter in print()?
10. How can we assess an element from the list?
11. How do you modify an existing item in a list

Long Questions

1. Explain the different types of decision-making statements in Python with examples for each.
2. Compare while loops and for loops.
3. Describe how to work with Python lists, including creation and modification. Provide code examples for adding items, removing items and accessing elements.

4. Explain the importance of testing and debugging in programming. Discuss different debugging techniques.
5. How do Python libraries enhance programming efficiency? Provide an example.
6. Write a Python program using a loop that prints all the odd numbers between 1 and 20.
7. Write a Python program that performs the following tasks:
    (a) Generates a list of 5 random numbers between 1 and 20.
    (b) Uses the statistics library to calculate and print the mean of the generated numbers.
8. Write a Python program that performs the following operations:
    (a) Create a list of popular Pakistani dishes: ["Biryani", "Nihari", "Karahi", "Seekh Kebabs"].
    (b) Add a new dish "Quorma" to the list.
    (c) Change "Karahi" to "Chicken Karahi".
    (d) Remove "Nihari" from the list.
9. Write a code to print stars (asterisk) in rectangular pattern using nested loops with three rows and five columns.
10. Write the output of the following code

```
print("=== Multiplication Table (1-5) ===")
for i in range(2, 4):          # Outer: 1 to 5
    for j in range(1, 11):     # Inner: 1 to 10
        print(f"{i} × {j} = {i*j:2d}", end="  ")
    print()  # New line after each table
```

UNIT 5

Introduction to Data Science

Student Learning Outcomes

By the end of this chapter, you will be able to:

- Explain the concept and importance of data science in everyday life.
- Describe each step in the Data Science Life Cycle (DSLC).
- Identify and define real-world problems that can be solved using data.
- Collect data using surveys, online forms, and observational techniques.
- Perform data cleaning and validation to improve accuracy.
- Use simple **SQL queries** (SELECT, INSERT) to retrieve and store data from databases.
- Analyze data using basic algorithms (sorting, classification, clustering).
- Create bar charts and dashboards to visually represent data insights.
- Communicate results clearly using charts and written summaries.

Introduction

Data Science is the process of using data to understand problems, discover patterns, and make smart decisions. In the current digital era, data is everywhere i.e., from surveys to mobile apps, social media, online shopping, and health services. Data Science helps us turn this raw data into useful information.

Whether it is predicting weather, understanding interests of students, or improving healthcare, Data Science is transforming the way we live, learn, and work. Organizations across the whole world use data to improve decision-making, save costs, and create better services.

This chapter introduces the Data Science Life Cycle (DSLC), which includes understanding the problem, collecting and cleaning data, analyzing and interpreting data, and finally communicating the results using charts and visuals. Each step builds critical thinking and problem-solving skills.

Students will also learn how to collect, clean, analyze, interpret visualize data using simple tools like surveys, spreadsheets, and charts. The chapter includes real-world examples, and hands-on activities that help them apply Data Science in classroom and daily life.

5.1 Introduction to Data Science

We are living in a world full of data. Whether we are using a mobile phone, searching on Google, shopping online, or even walking with a fitness tracker, data is being collected. The question arises how is this data useful?

Data Science is the field where we use data to find patterns, answer questions, and help people or businesses make better decisions. It involves collecting, organizing, analyzing, and visualizing data.

5.1.1. Importance of Data in Decision Making

Data plays a big role in helping people and organizations make better decisions. Without data, decisions may be based on guesses or personal opinions which can lead to mistakes. For example, if the school administration desires to start a new sports club, they can first collect data on which sports students enjoy most. This helps them make a smart and fair choice.

5.1.2. Real-Life Applications of Data Science

Data Science is used in many aspects of our daily life often without our notice.

- **Social media:** Applications like Facebook, YouTube and Instagram use data to show us posts video we might like.
- **Online shopping:** E-commerce websites suggest products using data of our previous purchases.
- **Healthcare:** Hospitals use data science to predict diseases and manage patient care more effectively.
- **Transport:** Applications like Google Maps use traffic data to suggest the fastest routes.
- **Education:** In schools, data from student attendance and test results is used to improve learning strategies and track progress.

Data Science supports in making our lives smarter, easier, and more personalized from watching YouTube videos to receiving health tips or choosing the best route.

5.2 Overview of the Data Science Life Cycle

Data Science is not covered in a single step. It follows a series of steps that are helpful in solving a problem. This is what we call Data Science Life Cycle (DSLC). DSLC usually comprises of the following steps:

Step 1: Understanding the Problem

Step 2: Collecting Data
Step 3: Cleaning and Validating Data
Step 4: Analyzing Data
Step 5: Interpreting Data
Step 6: Visualizing and Communicating Results

5.2.1. Understanding the Problem

Before we start talking about collecting data, it is necessary to clearly understand the problem under consideration. Understanding the problem is the most important step in DSLC. In case, we do not understand the problem properly, we may collect the wrong data resulting in useless or wrong results. In this initial step, we need to ask questions (what, why, who) such as:

- What are we trying to find out?
- Why is this problem important?
- Who will use the results?

The following sub-sections cover step 1 of DSLC in more detail.

Identifying and Defining a Problem

First, we need to **identify a problem** that we want to solve using data. The problem can be related to our school, home, city, or any area of life where decisions are made based on information. Once a problem is identified, then we must **define it clearly and simply**.

Example: Instead of saying, ***"students are not happy,"*** we can define the problem in a better way as ***"Which school activities do students enjoy the most?"***

When defining a problem, we need to consider the following: (i) Use clear and simple words, (ii) focus on one specific question, and (iii) avoid unclear or general terms.



TIDBIT

A good problem is one that can be answered by collecting and analyzing data.

Setting Clear Objectives

Once we have identified and defined the problem, we must decide **what we want to achieve**. These are called our **objectives** which help in guiding the rest of the data science process. They state (i) what kind of data we need, (ii) what questions

we will ask, and (iii) what results we are looking for.

Example: For the problem stated in above "Which school activities do students enjoy the most?", we can formulate our objectives as follows:

- Find out how many students participate in each activity.
- Find out which activity is rated as most enjoyable by the students.
- Recommend which activities should be improved or replaced.



TIDBIT

Write your objectives like a checklist. This helps keep your project focused.

Examples: School, Health, Business related Problems and Objectives

This subsection covers a few realistic examples to understand the concept in a better way. Table 5.1 provides example problems and objectives related to different domains such as school, health and business.

Table 5.1: Example Problems and Objectives

Domain	Example Problem	Objectives
School	*Many students arrive late at school*	Find out how many students arrive late in month.
		Identify reasons for being late (e.g., transport, distance, emergency, weather issues).
		Suggest solutions to reduce late arrivals.
Health	*Many students are not drinking enough water during school hours.*	Collect data on how much water students drink daily.
		Find out reasons for low water intake.
		Recommend steps to promote healthy habits.
Business	*A school canteen wants to know which snacks students like the most.*	Survey students on their favorite snacks.
		Compare popularity and sales data.
		Help the canteen manager make smart ordering decisions.

REMEMBER

1. The first step in any data science project is to understand the problem.
2. We must define the problem clearly and set specific goals.
3. Good problems are solvable using data.
4. Clear objectives help in collecting the right data and reaching useful results.

5.2.2.Collecting Data

Once we understand the problem and set clear goals, the next step in DSLC is to collect the required data. Data is like raw material, and we must gather the right kind of data before we can clean, store, or analyze it.

In Step 1, we asked questions like ***"Which school activities do students enjoy the most?"*** or ***"Why are students arriving late to school?"***

Step 2 is about collecting information that can help us answer these questions.

We will explore:

(i) data collection methods

(ii) where to collect data from (sources)

(iii) How to make sure it is useful and fair (data quality)

(i) Methods of Data Collection: There are many ways to collect data, depending on the problem and available tools. Let us discuss the most common ones used in schools, homes, and businesses.

- **Surveys (Online/paper based):** Surveys are the easiest and most common method. We ask people questions and record their answers. **For example:** To find out which sports students enjoy, we can create a survey with questions such as (i) Which sport do you like the most? (ii) How often do you play it? (iii) What time of day do you prefer to play?

DO YOU KNOW?

Surveys can be done using Google Forms (online), and Printed **paper forms** (manual).

- **APIs and Online Sources**: Application Programmer Interface (APIs) allow software collect data from websites or applications (apps in short) automatically. **For example**, A school may use an attendance API to pull daily attendance data into a report (use more by developers).
- **IoT and Sensor Data:** Internet of Things (IoT) devices like smart watches,

water dispensers, or temperature sensors can automatically collect data. **For example,** To study water drinking habits, schools can place smart water meters near fountains.



TIDBIT

IoT devices are useful in Health, safety, and environmental monitoring.

- **Web Scraping:** It is a process of programmatically collecting public data from websites. **For example**, A business student may scrape extract prices of snacks from online stores to compare them with school canteen prices.



TIDBIT

Web scraping should only be done with teacher for websites that allow it.

(ii) Choosing the Right Data Source

Not all data is useful. We must choose **reliable, accurate, and relevant** sources. When selecting a data source, ask a few questions like (i) Is this data from a trusted place? (ii) Is it recent or outdated? and (iii) Does it match the problem I want to solve?

Example: If we want to know favorite snacks of students in 2025, we should avoid using a survey done in 2020.



TIDBIT

Always collect fresh and problem-related data.

(iii) Assessing Data Quality and Bias

Data quality means the data is (i) Complete (no missing values), (ii) Correct (no mistakes), and (iii) Relevant (matches our problem).

Bias means the data only shows part of the truth and misses the full picture. For example, if we only survey boys about their favorite sports, we will miss the opinions of girls. The results will be unfair or biased toward boys.

Remember the following to avoid bias:

- Ask from a diverse group of people.
- Keep questions neutral and not leading.
- Review answers for strange patterns.



ACTIVITY

Designing a Simple Survey

Objective: Learn how to collect and store data by creating a short survey about free time habits of your classmates.

Instructions:

1. **Design Your Survey:** Think of **3 to 5 questions** to ask your classmates.
2. **Prepare to Collect Responses:** Choose a method (e.g., paper based or Google online form)
3. **Collect Data:** Ask at least 5 classmates to answer your survey and write down their responses clearly.
4. **Store the Data:** (i) If you are using paper, keep it neat and organized in columns. (ii) If using digital tools, save the file with a clear name like ***FreeTime_YourName.xlsx***

Optionally, show your teacher or classmates how you collected and stored the data.



REMEMBER

1. After defining a problem, we need to collect data that can help in solving the problem.
2. There are different methods of data collection such as surveys, APIs, sensors, and web scraping.
3. We should always check for data quality and fairness (bias) before moving to the next step.
4. Data that is incomplete or biased can lead to wrong decisions.

5.2.3. Cleaning and Validating Data

After collecting data, the next important step is to clean and validate it. Raw data is often messy. It may contain errors, missing values, or even wrong entries. If we use this data without checking it, our analysis will give wrong results. Remember that earlier we collected data using surveys and other methods. Now, we need to check if that data is correct, complete, and ready for use. This step ensures that the data we analyze later is trustworthy.

Raw Data

Raw data is the original information we collect from people, sensors, websites, or apps. It has not been checked or cleaned yet. For example, assume that 20 students filled your survey about favorite sports. Some entries may look like as:



Table 5.2: Raw Data about Favorite Sport of Students

ID.	Name	Favorite Sport	Frequency per week	Study Time of Day
1	Ahmad	cricket	Daily	Evening
2	Ali		Weekly	Afternoon
3	Sara	Footbal	Daily	
4	Sana	Football	Daliy	Evening

Table 5.2 contains:
(i) spelling errors i.e., Football (ID 3), Daliy (IDddd 4)
(ii) missing entries (ID 2 & 3)
(iii) Mixed Capitalizations i.e., cricket (ID 1), Football (ID 4). Such kind of data needs cleaning.

Common Data Errors (Missing, Duplicate, Outliers)

Table 5.3 shows some common issues found in raw data.

Table 5.3: Common Issues in Raw Data

ID.	Error Type	Example
1	Missing data	Blank cells (e.g., no sport name or study time of day)
2	Duplicate entries	Same student entered twice
3	Inconsistent formats	"Cricket", "cricket", "CRICKET"
4	Spelling mistakes	"Footbal", "Daliy"
5	Outliers (strange values)	Frequency = 1000 times/week

Data Validation Techniques

Data validation means checking if the data (i) Follows correct rules (e.g., only valid sport names), (ii) Has the right type of values (e.g., numbers where numbers are expected), (iii) Is not fake or silly.

Example Validation Rule: Only allow one of these values for "Favorite Sport": Football, Cricket, Badminton, Hockey.

Validation in Surveys: For this purpose, consider the following:

- Use multiple choice instead of blank text boxes.

- Make important questions as required, so no one skips them.
- Use dropdowns or checkboxes to reduce spelling mistakes.



TIDBIT

Tools like **Google Sheets**, **Excel**, provides rules that can help validate data.

Data Cleaning Methods

Once we find mistakes, we need to fix or remove them. This is called data cleaning.

The following are a few basic steps to clean data:

- Fix spelling errors: "Footbal" → "Football"
- Standardize formats: "CRICKET" → "Cricket"
- Remove duplicates (redundancy): Delete repeated entries for the same student.
- Handle missing values: Ask the person again OR remove the row.
- Fix invalid data: If frequency is written as "ten hundred", fix it or remove it.

Why Cleaning Data Matters

Suppose that we skip this step and we analyze data that includes:

- Misspelled sports ("Cricket")
- Missing Times of the Day / Study time.
- Duplicates

Our result might say: "Football: 3 votes, Cricket: 1 vote, Cricket: 2 votes" → This gives the wrong picture, because “Cricket” and “Cricket” are the same sport!

In summary, we can say that clean data leads to smart decisions. Dirty data leads to incorrect result.

5.2.4. Analyzing Data

Now that our data is collected, and cleaned properly, we can move on to the analysis steps. This is where we look at the data to find patterns, trends, and answers to the questions we asked in Step 1.

Suppose that you have a clean and organized table of data, for example, student responses about their favorite sport or study times. Now you may ask question like “what does this data tell us?” This step helps us make decisions based on

what we discover in the data.

What Does it mean to Analyze Data

Analyzing data means carefully examining data to (i) Summarize it (e.g., how many students chose each option), (ii) Compare different groups (e.g., boys vs girls, morning vs evening), (iii) Find patterns (e.g., most students prefer football and play in the evening).

You may use measures like counts (how many?), averages, frequency (how often?) etc.

Using Simple Tools for Analysis

You can start analyzing data with simple tools such as Google Sheets, Microsoft Excel or manual writing in a notebook

Example: You surveyed 20 students. Table 5.4 shows what you may have found.

Table 5.4: Data Derived from Student Survey

Favorite Sport	Frequency
Cricket	10
Football	5
Hockey	4
Badminton	1

From this data, we can observe that ***most students prefer cricket***.

ACTIVITY Data Cleaning

Objective: Learn how to identify and correct common data entry mistakes (such as spelling errors, missing values, and inconsistent formatting).

Instructions:

1. Review the Messy Table 5.2

Table 4: Raw Data about Favorite Sport of Students (with Messy Data)

Sr. No.	Name	Favorite Sport	Frequency	Time of Day
1	Sadiq	Cricket	Daliy	Evening
2	Noor	Cricket	Weekly	
3	Maryam	Football		morning
4	Zara	Football	Daily	Afternoon

2. Work in pairs: Sit with a partner and study the table together.

3. Spot and List Errors:

Find **at least 5 errors**. These may include:

- Spelling mistakes (e.g., *Cricket*, *Daliy*)
- Missing data (e.g., blank cells)
- Inconsistent formatting (e.g., *Football* vs *football*, *morning* not capitalized)

4. Correct the Errors:
(i) Discuss how to fix each error logically and fairly.
(ii) Rewrite the **cleaned table** neatly in your notebook.
5. Class Sharing (optional):
(i) Compare your cleaned table with another group.
(ii) See if they found the same errors or different ones.

REMEMBER

1. Data analysis helps us find patterns and make decisions.
2. We use counts and comparisons to understand data.
3. Even simple tools like tables or spreadsheets can help us analyze data.

ACTIVITY **Data Analysis**

Objective: Learn how to analyze data by identifying trends and drawing basic conclusions from a frequency table.
Instructions:

- Use your own survey data (if available) or the sample data given in Table 5.5 below:

Table 5.5: Sample Derived Data from Survey Data

Sport	Frequency or No. of Students
Cricket	7
Football	6
Hockey	5
Badminton	2

- Carefully read the table showing how many students like each sport.
- Identify: (i) The **most popular** sport, (ii) The **least popular** sport
- Write **one reason** why each sport might be most or least popular in your class.
- Present your findings in your notebook using simple sentences.

5.2.5.Interpreting Data

This step is to finding meaning from data (i.e., identifying trends and patterns). After summarizing the data, we may ask ourselves:

- What do the numbers tell us?
- Why are some options more popular?
- Are there any surprises or patterns?

Example: If we find that most late students live far from the school, we can recommend providing transport support as a solution.

5.2.6. Visualizing and Communicating Results

After analyzing the data, the final step is to share the results in a way that others can understand easily. People usually understand pictures better than numbers, which is why we use data visualizations.

Remember that in previous step, we found that cricket was the most popular sport. In this step, we will create a chart that shows this clearly. This makes it easy to present our findings to classmates, teachers, or others.

Importance of Data Visualization

Data Visualization is important because it

(i) makes data easier to understand,

(ii) shows patterns at a glance,

(iii) helps share ideas clearly in presentations or reports.

Example: Instead of saying "10 students like cricket," we can show it in a bar chart.

Common Types of Charts and Graphs (Bar, Pie, Line, etc.)

Always choose a chart that **matches your data**. Table 5.6 shows the commonly used types of charts, its use case with examples.

Table 5.6: Common Types of Charts

Chart / Bar Type		Used For	Example
Bar Chart		Comparing groups	Number of students who like each sport
Pie Chart		Showing parts of a whole	Percentage of favorite snacks
Line Chart		Showing changes over time	Number of students arriving late each week

TIDBIT

Keep your charts simple, clean, and properly labeled.

Communicating Results

Once the chart is ready, explain your findings. You can write 2–3 lines:
"Most students (10) like cricket, followed by football (6). Very few students like badminton (1)."

TIDBITS

Use visuals and words together to tell a **data story** (clear, short, and useful).

REMEMBER

1. Visualizing **data** means using charts or graphs to show findings.
2. It makes data easier to understand and share.
3. Use the right chart for your data type.
4. Always **label and explain** your visuals.

5.3 Tools for Visualization

Once you know which type of chart to use, the next step is to create it using tools. In Step 5, we learned how charts make it easier to understand and present data. In this section, you will learn **how to actually make those charts** using software. Let us explore the most useful and beginner-friendly tools.

5.3.1. Creating Charts in Spreadsheets (Excel/Google Sheets)

Google Sheets and **Microsoft Excel** are simple yet powerful tools to create charts and graphs from tables.

Steps to Create a Chart in Google Sheets:

i. Enter your data into rows and columns.
ii. Select the data range.
iii. Click on the "Insert" menu → choose "Chart."
iv. Pick the chart type (bar, pie, line).
v. Customize the chart with a title, label, and color.

Example: Assume that you have data (refer to Table 5.5) in Google Sheets. You are encouraged to use these steps to turn Table 5.5 data into a bar chart.

We make use of spreadsheets because they are (i) Easy to use, (ii) No installation needed (for Google Sheets), (iii) Good for small to medium-sized projects Figure 5.1 shows a simple bar graph corresponding to our Table 5.5.

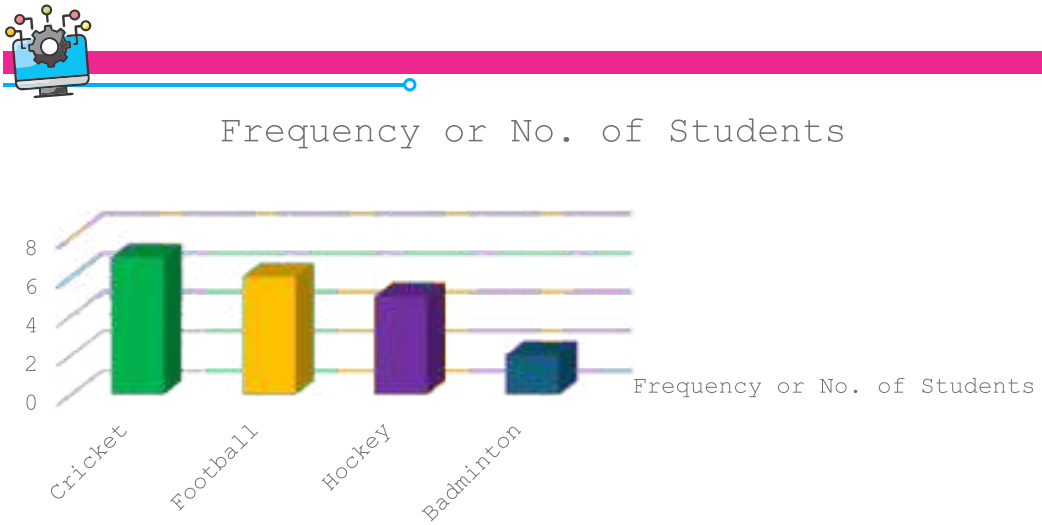


Figure 5.1: Favorite Sport of Students (Based on Data of Table 5.5)

ACTIVITY

Drawing a Bar Cart, Pie chart and Line Chart from Survey Data given in table 5.4

Objective: Learn how to convert a frequency table into a bar chart to visually represent survey data.

Instructions:

- Use the sample data given in **Table 5.4**.
- Draw a **bar chart, pie chart and line chart** in your notebook.
- Present your findings in your notebook using simple sentences.

REMEMBER

1. Google **Sheets** and **Excel** are easy tools for creating charts.
2. A **dashboard** shows multiple visuals in one place to tell a clear data story.

Note: The standard DSLC consists of 6 main steps, however, in some real-world applications/ scenarios, some additional steps may be added. One such step can be **Storing and Organizing Data**, especially in large projects or team settings where databases and structured storage are important. These are usually managed through systems like DBMS and SQL.

5.4 Databases and Storing Data

After cleaning a messy survey table, there is a need to store the cleaned data in a proper format (either in a spreadsheet or a database), so that it is organized, searchable, and ready to analyze. Let us explore how and where to store data. There are two main ways to store structured data i.e., (i) Spreadsheets, and (ii) Databases. Difference between both is provided in Table 5.7.

Table 5.7: Spreadsheets vs Databases

Method	Description	Example Tool
Spreadsheets	A grid of rows and columns for small datasets	Excel, Google Sheets
Databases	A structured system to store and manage large data in tables	MySQL, SQLite

TIDBIT

Spreadsheets are good for small projects. Databases are better for big or multi-user data systems.

5.4.1 What is Database?

A Database is an organized collection of data. It stores data in the form of tables. Each table is like a mini spreadsheet with rows and columns. Databases help in storing, retrieving, and updating data efficiently. Table 5.8 breaks down database related terms.

Table 5.8: Database Terms: Meanings and Examples

Term	Meaning	Example
Table	A group of related data	Sports Survey
Record	A single entry (row) in the table	Ali's full row of data
Field	A specific piece of information	Name, Sport, Frequency

ACTIVITY

Data Organization

Objective: Learn how to organize cleaned data into a well-structured table using consistent formatting.

Instructions:

- Use the cleaned survey data you collected earlier.

- Create a table in your notebook with the following column headings: Roll No, Name, Favorite Sport, Time of Day, Frequency
- Fill in at least 5 rows of data.
- Keep your data aligned properly in columns.
- Ensure all entries are spelled correctly, capitalized properly, and free from errors.

A Database Management System (DBMS) is software used to create, manage, and interact with databases. Examples include MySQL, SQLite, and Microsoft Access.

To work with data in a database, we use a language called Structured Query Language (SQL). SQL helps us to add, update, delete, or retrieve data using simple commands. Some basic SQL queries are as follows:

- SELECT * FROM Sports: This simple query shows all the data from the table named *Sports*.
- SELECT Name FROM Sports WHERE Favorite_Sport = 'Cricket': This query shows names of students present in Sports table who like cricket.
- INSERT INTO Sports (Name, Favorite_Sport) VALUES ('Ali', 'Football'): This query adds a new entry.

These basic SQL statements help in managing and retrieving specific data from large databases efficiently.

REMEMBER

1. Data should be stored in a **structured** way for easy access and analysis.
2. **Spreadsheets** are useful for small datasets.
3. **Databases** are used for bigger and more complex data
4. **A table** stores data using **records (rows)** and **fields (columns)**.
5. Organized data reduces duplication errors and makes analysis easier.

Summary

- **Spreadsheets**: Grid of rows and columns for small datasets (Excel, Google Sheets)
- **Database**: Organized collection of data stored in tables
- **Table**: A group of related data (like a mini-spreadsheet)
- **Record**: A single entry (row) in a table
- **Field**: A specific piece of information (column) in a table
- **SQL (Structured Query Language)**: Language used to work with data in relational databases

EXERCISE

Multiple Choice Questions:

1. **What is the first step in the Data Science Life Cycle (DSLC)?**

   (a) Collecting Data (b) Visualizing Data
   (c) Understanding the Problem (d) Analyzing the Data

2. **Which method is best for collecting opinions from many students?**

   (a) IoT Devices (b) Survey
   (c) Web Scraping (d) Database

3. **What is "bias" in data?**

   (a) Very large data size (b) Unfair or one-sided data
   (c) Well-cleaned data (d) High-speed data

4. **Which tool is ideal for beginners to create charts?**

   (a) Tableau (b) Python
   (c) Google Sheets (d) Hadoop

5. **A relational store information in the form of----------.**

   (a) Slides (b) Tables
   (c) Blocks (d) Layers

6. **Which of the following charts is best for comparing groups?**

(a) Pie Chart (b) Bar Chart

(c) Line Chart (d) Area Chart

7. **What is the main goal of analyzing data?**

(a) Delete old data (b) Decorate data

(c) Find patterns and make decisions (d) Upload to the internet

8. **Which of the following tools is an advanced data visualization platform?**

(a) MS Paint (b) Google Data Studio

(c) Notepad (d) WhatsApp

9. **What should be done before storing data?**

(a) Add animations (b) Clean and validate it

(c) Encrypt it (d) Translate it

10. **Which one is an example of a data source?**

(a) YouTube (b) Survey form

(c) Whiteboard (d) Slide show

Short Questions

1. What is the importance of understanding a problem before collecting data?
2. Name any two methods of collecting data and give one example of each.
3. What do we mean by "cleaning" data?
4. Why is it important to validate data before using it?
5. What is the purpose of visualizing data using charts?
6. How does bias in data affect results?
7. Write two steps you will follow to clean a messy survey.
8. Name any two tools used for creating charts.

Long Questions:

1. Explain the six steps of the Data Science Life Cycle (DSLC) with one line each.
2. Give a detailed comparison between Bar Chart, Pie Chart, and Line Chart with examples.
3. Differentiate between DBMS and SQL
4. Write a Query in SQL to get all data of a student who's roll_number is 15 from student table.

UNIT 6

Introduction Artificial Intelligence (AI), and Machine Learning (ML)

Student Learning Outcomes

By the end of this chapter, you will be able to:

- Define artificial intelligence (AI), and machine learning (ML).
- Differentiate between AI, and ML, understanding their unique characteristics and applications.
- Understand the concept of supervised learning and its application to teach systems to make predictions.
- Comprehend unsupervised learning, used for clustering and pattern recognition.
- Understand different algorithms and choose relevant algorithms based on specific business problems. This includes simple linear regression.
- Familiarity with key model performance metrics accuracy.
- Understand where AI and ML are applied in daily life, business, and education.
- Discuss bias, privacy, and responsibility when using AI.

Introduction

In the present digital era, technology is not just about machines, it is about smart systems that can think, learn, and help us in making better decisions. This chapter introduces the exciting fields of Artificial Intelligence (AI), and Machine Learning (ML). These are powerful technologies behind many commonly used things such as voice assistants like google assistant, ChatGPT, video recommendations on YouTube, and even predicting the weather. Understanding these fields helps you see how computers are becoming more intelligent and helpful in solving real-life problems.

In this chapter, Students will explore how data is used to train machines to recognize patterns, make predictions, and group information just like a human brain, but more quickly and accurately. how these technologies are shaping the future and how they can play a part in this paradigm shift.

6.1 AI, ML and Overview

In the modern world, huge amounts of data are being collected every second (more than humans can handle alone). This is where AI and ML come in. These technologies learn from data and help machines think, decide, and improve (just like humans) but faster and more consistently.

6.1.1 What is AI?

AI means teaching machines to do tasks that usually require human intelligence such as understanding speech, recognizing images, or making decisions.

Definition: AI is the ability of a machine or computer program to perform tasks that normally require human thinking and ability.

Real Life Examples:

- Urdu speech-to-text apps used in educational institutes including schools
- Chatbots used by banks like HBL and JazzCash for customer support
- Traffic control systems in big cities such as one introduced by Safe City Authority in Lahore.

6.1.2 What is ML?

Machine Learning is a part of AI. It means training a machine to learn from data and make decisions on its own without being programmed every time.

Definition: ML is a method where computers learn from past data to predict or decide something.

Examples:

- Netflix or YouTube recommend videos based on watching history of users.
- A weather app uses past temperatures to predict weather for next day.

6.1.3 Differences between AI and ML

The Table 6.1 shows the differences between AI and ML.

Table 6.1: Differences between AI and ML

Feature	AI	ML
Definition	Making machines smart enough to mimic human behavior	Teaching machines to learn from data and improve

Main Goal	To perform smart tasks like humans	To make predictions or decisions from data
Needs Data?	Sometimes (uses data to act smartly)	Always (needs data to learn)
Human Like Behavior?	Yes (like speaking, seeing, or understanding)	No (just learns patterns)
Real Life Example	Chatbot on JazzCash that replies to customers	App that predicts electricity usage next week

TIDBITS

- Think of AI as the brain, and ML as the experience that teaches brain how to improve.
- AI is the broader concept of machines acting smart, while ML is when machines learn from data to make predictions or decisions.

6.2 Machine Learning Types

In the previous section, you learned that ML helps machines learn from data to make predictions or decisions. Just like students can learn with the help of a teacher or by exploring on their own, machines can also learn in two main ways, i.e., Supervised Learning and Unsupervised Learning.

6.2.1 Supervised Learning

In supervised learning, the machine is given data with correct answers **(labels)**. It learns by comparing its own guesses with the correct answers and its learning improves over the time.

Definition: Supervised learning is when the machine is trained using labeled data, where both instance and correct output are provided.

Example 1: Suppose you are training a machine to predict whether a fruit is an apple or a mango. You show the machine many Apples and Mangoes and label it as that

- "This is an Apple"
- "This is a Mango"

The computer learns from these examples. Later, it can predict with high accuracy when shown a picture of apple and mangoes such pictures.

Example 2: A system can be trained based on marks and attendance of students to predict who might fail.

6.2.2 Unsupervised Learning

In unsupervised learning, the machine is given only data without any labels. It has to find patterns or group similar things by itself.

Definition: Unsupervised learning is when the machine finds patterns or groups in unlabeled data without being told what the correct answer is.

This helps the store in marketing products in a better way, even without knowing the exact customer types.

Example 2: A school wants to divide students into study groups. The system checks their interests and subjects and forms groups just based on similarities and without knowing which group is "best".

6.2.3 Supervised vs Unsupervised Learning

The comparison between both types of learning is provided in Table 6.3.

Table 6.3: Supervised vs Unsupervised Learning

Feature	Supervised Learning	Unsupervised Learning
Labeled Data?	Yes (machine is told the correct answers)	No (machine finds patterns on its own)
Main Task	Predict or classify	Group or cluster similar data
Example	Predict student results using marks and study hours	Group customers based on shopping behavior
Uses in School	Identify students who need help	Make automatic student groups for projects

ACTIVITY

Discuss and Match (Supervised or Unsupervised Learning)

Objective: Understand the difference between **Supervised** and **Unsupervised** Machine Learning by identifying the correct type for real-life situations.

Instructions:

- Read each situation below carefully (refer to Table 6.4)

Table 6.4: Match as **Supervised** (S) or **Unsupervised (U) Learning**

Sr. No	Situation Statement	S or U
1	Predicting if a student will pass or fail	
2	Grouping students based on subject preferences	
3	Identifying spam emails from inbox	
4	Dividing products into types based on features	

- In your notebook, write the correct ML type for each: whether each situation is an application of **supervised or unsupervised learning**.
- Write your answers in your notebook with a brief reason for each.
- Discuss your answers with a classmate and groups
- Explain your reasoning during class discussion.

TIDBIT

Think of **Supervised Learning** as learning with a teacher, and **Unsupervised Learning** as exploring on your own.

REMEMBER

1. **Supervised Learning** uses labeled data to **predict or classify**.
2. **Unsupervised Learning** uses unlabeled data to **find hidden patterns**.
3. Both are used in schools, businesses, apps, and government systems in Pakistan.
4. If the system is learning from examples with correct answers, it is supervised. If it is grouping without given answers, it is unsupervised.

6.3 Practical Applications of AI/ML

Now that you understand what AI and ML are, and how machines learn, now it is time to explore where these technologies are being used around us. We use these technologies everyday seamlessly.

6.3.1 Smart Assistants and Recommendations

AI powers many smart tools and apps that help make our lives easier.

Smart Assistants: Smart assistants like Google Assistant, Siri, use AI to understand voice commands, translate languages, set reminders, and answer questions.

Example: Say "Kal 8 bajay mujhe jagana" to Google Assistant in Urdu, it will set an alarm using speech recognition and natural language processing techniques, both powered by AI.

Recommendation systems: These systems suggest what you might like, based on your past activities.

Examples:

- YouTube recommends videos after you watch a few clips.
- Daraz suggests clothes based on your browsing history.
- Instagram shows videos similar to what you have watched before.

These apps use **ML** to analyze what you like and improve their suggestions over time.

TIDBIT

If the app makes suggestions, understands your voice, answers you, or learns your habits.

ACTIVITY

AI in Our Daily Life

Objective: Recognize how AI and ML are used in everyday tools and applications.

Instructions:

- List **3 Apps** you have used during previous week that may use AI. For example, YouTube, Google Maps, ChatGPT, Snapchat.
- With a class partner, discuss **what the AI does** in each app. Does it suggest videos? Does it respond to voice commands? Does it help you search faster?
- Write your examples and the AI Role in your notebook.

6.3.2 Business Use Cases Using Customer Analysis and Chatbots

AI help businesses understand customers, answer questions, and improve their services.

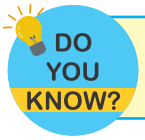
Customer Analysis: Companies collect data about customers to find:

- Which product is best to sell.

Example: A grocery delivery app like **Cheetay** uses ML to suggest items you often buy, and predicts which items are needed in which city.

Chatbots and Automated Help: Many Pakistani banks, telecoms, and e-commerce sites now use chatbots which are smart systems that can answer

questions without needing a human agent.
These chatbots use AI to **understand typed questions** and respond automatically.



AI helps businesses **work faster**, **save money**, and **keep customers happy** using the power of data.

6.3.3 Bias, Privacy, and Responsibility

While AI are powerful, they must be used **responsibly**.
Bias in AI: Sometimes, machines **learn unfair patterns** from the data they are trained on. This is called **bias**. **For example**, if an AI system is trained only on English or male voices, it may not work well with Urdu speakers or female voices.



Always use diverse and balanced data to reduce bias.

Privacy and Data Use: AI systems often use **personal data** (like name, location, or choices) which raises privacy concerns. For example, if an app tracks your movements or listens to your microphone without permission, it can be misused.
Best Practice:

- Ask for user permission
- Protect data using encryption
- Do not collect more than needed

Responsibility of Humans: Humans must use AI wisely as it is just a tool. If an AI powered system makes a wrong decision, **the responsibility lies with the humans who created or used it**. For e**xample**, if a facial recognition system wrongly accuses system should be fixed and detect.



- AI is used in smart assistants, shopping apps, and chatbots.
- Businesses use them for customer service and sales analysis.
- We must always consider ethics: fairness, privacy, and responsibility.

6.3 Supervised and Unsupervised Algorithms

As stated in the previous Chapter, Data science uses various algorithms to find patterns, make predictions, or analyze large data. Choosing the right algorithm depends on the type of data and the kind of problem you are trying to solve. Some common data science algorithms are:

- **Linear Regression:** Predicts future values based on a straight-line relationship (e.g., predicting student scores from study hours).
- **Decision Trees:** Uses a tree-like model to make decisions based on yes/no questions (e.g., predicting if a customer will buy or not).
- **K-Means (Clustering):** Groups similar items together (e.g., grouping students with similar interests).

These algorithms help organizations solve real-world problems such as predicting exam results, grouping customers for better marketing and identifying patterns in fraud or health data.

TIDBIT

Think of an algorithm like a recipe in the kitchen. It tells the computer what ingredients (data) to use and what steps to follow to get the final dish (prediction or group).

6.4.1 Linear Regression

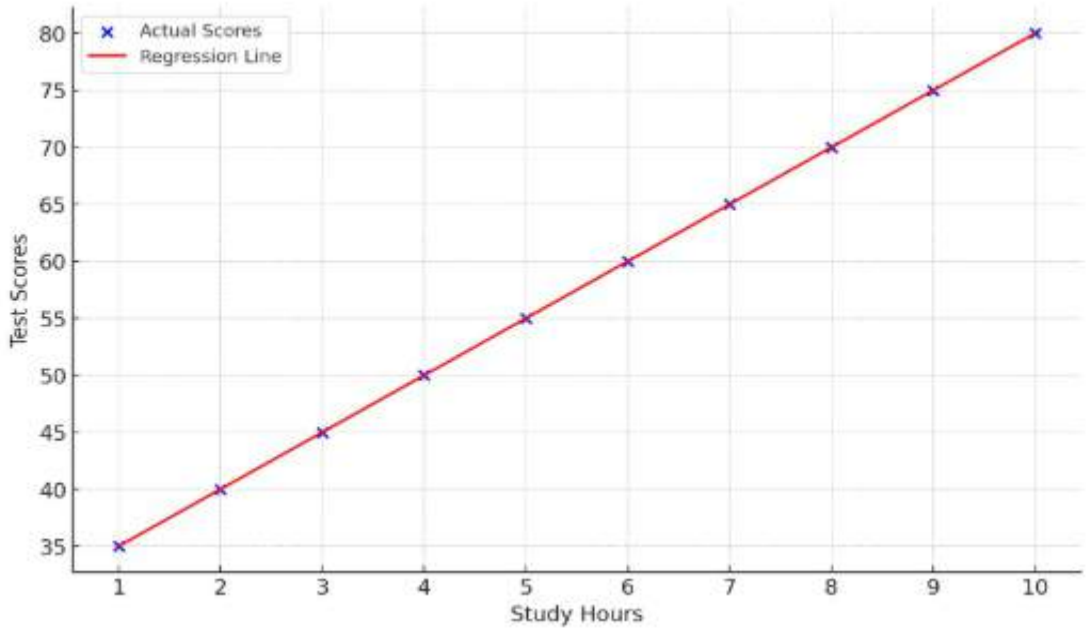
Simple Linear Regression algorithm helps us predict a value based on the relationship between two things. For example, a teacher wants to predict how well students will perform in a test based on the number of hours they studied. Here, study hours is taken as Input (X) while test score is used as Output (Y). Linear regression finds a line that best fits the data and helps predict marks if students studied for, say, 6 hours. Predicting sales, temperature, or student performance based on past data are among the few use cases of linear regression.

Scenario: A teacher wants to predict student test scores based on their study hours using data provided in Table 6.5. Linear Regression draws a best-fit straight line to predict future values as shown in Figure 6.1. In this graph, blue dots are actual student scores while blue line shows the predicted trend. If a student studies for 6.5 hours, the model can predict their score using the line.

Table 6.5: Student Data

Study Hours	Test Scores
01	35
02	40
03	45
04	50
05	55
06	60
07	65
08	70
09	75
10	80

Table 6.5: Study Hours vs Test Score



Linear Regression: Study Hours vs Test Score

DO YOU KNOW?

- Simple linear regression uses just one independent variable (input) to predict one outcome. However, in real life, outcomes usually depend on many things.
- Multiple linear regression helps the computer understand more complex patterns and improve prediction accuracy.

6.5 Evaluating Model Performance

Once a ML model is trained, (like predicting who will buy a product or whether a message is spam), then we are required to measure the performance of the model. This helps in deciding if the model is accurate and reliable. In this section, we will explore four important performance metrics with simple definitions and examples.

6.5.1 Confusion Matrix

A **confusion matrix** is a simple table that shows how many predictions/ classifications were correct or incorrect.

Table 6.6 shows a simple confusion matrix considering whether a student will pass or fail (classification problem).

Table 6.6: Confusion Matric

	Predicted: pass	**Predicted: fail**
Actual: pass	True Positive (TP)	False Negative (FN)
Actual: fail	False Positive (FP)	Ture Negative (TN)

- **TP =** Model predicted pass, and the student was pass in actual as well.
- **TN =** Model predicted fail and the student failed in actual as well.
- **FP** = Model predicted pass, but the student failed in actual.
- **FN** = Model predicted fail and the student was pass in actual.

The confusion matrix helps in understand whether a model is making classification or prediction mistakes and helps in calculating the accuracy (an important measure) of the model.

Accuracy

Accuracy (*A*) tells us how many predictions were correct out of all predictions made.

It is calculated as follows: $Accuracy = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$

Let us assume that we have data for 15 students. In actual, 8 students were pass while 7 failed. The model produced the following results:

- TP = Correctly predicted pass = 7
- TN = Correctly predicted fail = 5

- FP = Wrongly predicted pass = 2
- FN = Wrongly predicted fail = 1

Using the above formulas and model results, we calculate accuracy as follows:

$$A = \frac{7+5}{7+5+2+1} = \frac{12}{15} = 0.80\ or\ 80\%$$

Summary

- AI enables machines to perform smart tasks. ML, a part of AI, allows machines to learn from data to make predictions or identify patterns without being explicitly programmed every time
- Supervised Learning (e.g., predicting student grades or customer purchases)
- Unsupervised Learning (e.g., grouping people by preferences or behavior)
- Linear Regression for predicting values
- Decision Trees for making decisions
- K-Means Clustering for finding patterns in data (overview)

EXERCISE

Multiple Choice Questions:

1.What is Artificial Intelligence (AI)?

(a)Solving math problems by hand

(b)Machines performing tasks like humans

(c)Sending emails using a computer

(d)Watching videos online

2.Which of the following is an example of AI?

(a)A chatbot replying to your questions

(b)Using a whiteboard

(c)A TV remote

(d)Typing on a keyboard



3.What is Machine Learning (ML)?

(a)Teaching humans to use machines

(b)Teaching machines to learn from data like human

(c)Learning computer hardware basics

(d)A machine without internet

4.Which app recommends videos using ML?

(a)Google Maps

(b)Microsoft Paint

(c)YouTube

(d)Windows Media Player

5.In supervised learning, the data is:

(a)Unorganized

(b)Without any answers

(c)Labeled with correct answers

(d)Only text-based

6.Which of the following is an example of unsupervised learning?

(a)Predicting student marks

(b)Identifying spam emails

(c)Marking quizzes

(d)Grouping students by interest without labels

7.Google Assistant is an example of:

(a)A game app

(b)A word processor

(c)A smart assistant using AI

(d)A calculator

8.What skill is essential for a data analyst?

(a)Painting

(b)Basic drawing

(c)Interpreting and visualizing data

(d)Flying drones

9.AI is mostly used for:

(a)Handwriting notes

(b)Making phone calls

(c)Charging phones

(d)Smart decision making and automation



10.A chatbot on JazzCash is an example of:

(a)Cloud computing (b)Data visualization

(c)AI in business (d)Unsupervised learning

Short Answer Questions

1. What is the difference between AI and ML?
2. Give one real-life example of AI used in Pakistan.
3. How does supervised learning work?
4. Define confusion matrix.
5. How can businesses use AI?
6. Give one example each of a smart assistant and a recommendation system.
7. What kind of data is used in unsupervised learning?

Long Answer Questions

1. Explain supervised and unsupervised learning with suitable examples.
2. Explain how model performance is evaluated in machine learning using the following metrics:
   a) Confusion Matrix
   b) Accuracy
   c) Precision
3. A model predicts if students will pass (1) or fail (0). Out of 20 students:
   True Positives (TP) = 8
   True Negatives (TN) = 6
   False Positives (FP) = 4
   False Negatives (FN) = 2

You are required to (a) fill the confusion matrix, and (b) calculate accuracy.

UNIT 7

Applications of AI

Student Learning Outcomes

By the end of this chapter, you will be able to:

- Explain how AI can be applied to specific applications in areas like NLP, Robotics, Speech Recognition.
- Conduct a technology recognition experiment where students interact with products based on NLP, Robotics, Speech Recognition.
- Discuss the benefits and limitations of NLP, Robotics, Speech Recognition, and others
- Show that there are instances where use of AI causes social injustices.
- Develop critical thinking skills to assess situations where AI has led to social injustices and identify potential areas of improvement.
- Effectively communicate the social implications of AI, particularly regarding injustices and ethical responsibilities, both in writing and verbally.
- Exhibit an awareness of ethical considerations in AI development and usage, as well as the potential consequences of overlooking these considerations.
- Analyze case studies or scenarios where AI has caused social injustices, and discuss the underlying factors contributing to these issues.

Introduction

Computer Science is not just about using computers, it is about solving problems, creating smart tools, and making life easier. One of the most important of modern computer science is Artificial Intelligence (AI).

Natural Language Processing (NLP) helps computers understand human language. Chatbots, text analysis tools, and automatic translation apps are all part of NLP. Speech technology powers voice assistants like Alexa and Google Assistant. Recommendation systems suggest audios, videos, or products based on what users like. Autonomous vehicles, such as self-driving cars, use AI to move safely without drivers. Robotics is used in factories, hospitals, and even space missions.

AI must be used fairly and safely. It is important to study ethics, so that AI does

not harm people or create social injustices. Some systems may show bias or make unfair decisions if they are not designed carefully. That's why students and future developers must learn about data privacy, accountability, and ethical rules for creating AI tools. This chapter will also teach how to think critically about AI, recognize both its benefits and risks, and understand how AI affects jobs, education, health, and society.

7.1 Introduction to Artificial Intelligence (AI)

Artificial Intelligence (AI) is a modern field of computer science that teaches machines how to "think" and "act" like humans. It helps computers solve problems, learn from data, and make smart decisions without being told what to do step by step. AI is used in many areas such as mobile apps, online shopping, health care, and transportation. For example, when you use a voice assistant like Google assistant or a chatbot on a website, you are using AI. It helps doctors find diseases, drivers use maps, and students find answers online.



Figure 7.1: AI helping people

TIDBIT

AI can now write stories, solve math problems, and even create art, just like a human!

ACTIVITY

"AI Around Us" Poster Task

Students will work in pairs to create a small poster showing how AI is used in different parts of society — such as hospitals, schools, or banks. They can draw, use magazine cutouts, or describe with words. Posters will be shared in class to understand AI's importance in different sectors.



Figure 7.2: AI helps doctors in the diagnosis of diseases

7.1.1 Impact of AI on Jobs and Work?

AI is changing the way people work in many fields. In factories, robots and machines now do jobs that were once done by human experts. These machines can work faster, longer, and with fewer mistakes. At the same time, AI is also creating new kinds of jobs. People are needed to design, test, and manage AI systems. Jobs like data scientist, AI developer, and machine learning engineer are growing fast.

ACTIVITY

"Before and AfterC AI" hart

Students will be given a worksheet with two columns: "Before AI" and "After AI." They will list jobs or tasks that humans experts used to do and how AI now helps with or replaces those jobs.

7.2 Data and AI

Data is the most important part of AI. AI learns from data, just like a student learns by reading books. If the data is large and correct, the AI system becomes smarter and gives better results. AI systems depend on data to learn and improve. When they receive a large amount of information from different sources, they can find patterns and learn how to make better choices.

DO YOU KNOW?

An AI trained with poor-quality data can give wrong answers or make unfair decisions.

7.2.1 Automatic Data Collection

Automatic data collection means gathering information without needing a person to enter it by hand. Many devices and applications can collect data on their own. For example, mobile phones collect location, usage, and search history automatically. Online websites also collect user clicks, views, and preferences.

ACTIVITY

"What Data Do Apps Collect?" Survey

Students will check 2–3 apps on their phone and note what kind of data the app collects (location, voice, clicks, search history).

With automatic collection, AI can adjust and respond to user needs quickly. It also saves time, reduces errors, and allows systems to work without needing human input all the time.

7.2.2 Types of Data: Quality and Quantity

In Artificial Intelligence (AI), the type of data used is very important. "Quantity refers to how much data is available". When an AI system receives a large number of examples, it becomes more familiar with different situations. For instance, if a machine sees many different images or texts, it becomes better at understanding new ones. "Quality means how useful and correct the data is". If the data has mistakes, missing values, or confusing labels, the AI may learn the wrong things. Therefore, both the amount and the accuracy of data play a big role in building strong AI systems.

DO YOU KNOW?

AI learns better with both types of data: big (quantity) and accurate (quality).

ACTIVITY

Clean vs. Dirty Data" Matching Task

Students are given two small data tables: one with clear (good quality) and one with missing or wrong details (bad quality). They match questions to each table and realize how difficult it is for AI to make good decisions with poor data.

7.3 Real-World Applications of AI

AI is now used in many areas of daily life. It helps businesses, schools, hospitals, and even homes become smarter and more efficient. These are real uses of AI that make work easier and faster.

7.3.1 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a part of AI that helps computers understand human language. It allows machines to read text, listen to speech, and give answers in a way people can understand. For example, when someone types a question in a search bar or talks to a voice assistant, NLP helps the AI system understand and reply correctly (as shown in Figure 7.3). In customer service, it helps answer common questions without human help.

TIDBIT

Chatbots on websites can answer your questions 24/7, all thanks to NLP.



Figure 7.3: Talking Chatbot helps people

7.3.2 Robotics in Everyday Life

Robotics is an important part of AI that involves machines designed to do tasks like humans. These machines are called robots. They can perform tasks like lifting heavy items, helping in surgeries, or cleaning floors intelligently. Robots

can help old or disabled people with simple daily tasks. Because robots do not get tired and can work with great accuracy.

DO YOU KNOW?

Robots are now used in some restaurants to serve food and clean tables.

7.3.3 Speech Technology and Voice Control

Speech technology is a type of AI that allows computers to listen and understand spoken words. It helps machines turn voice into text and respond with answers. Voice assistants like those used in phones or smart devices are examples of this technology. For example, in customer service, it helps answer calls and guide users without a human operator. In smart homes, people can control lights or devices just by speaking. In healthcare, it can help doctors record notes without typing. Speech technology makes it easier and faster for people to use machines with just their voice.

TIDBIT

Smartphones can now unlock using your voice , no need to touch anything.

DO YOU KNOW?

Some cars now respond to voice commands to turn on music, make calls, or give directions.

ACTIVITY

"Speak and See" Practice

Using a phone or computer, students try out voice-to-text tools (like Google Voice Typing). They speak a sentence and see if the tool types it correctly. Then they try speaking in different tones or speeds and compare the results.

7.3.4 Recommendation Systems

Recommendation systems are a part of AI that give users suggestions based on their likes and past actions. These systems study what a person has viewed, searched, or selected before, and then show similar options. For example, on an online shopping website, if someone looks at school bags, the AI may suggest other types of bags or related items. In education, a website can suggest helpful videos or articles based on a student's learning history. Recommendation

systems improve the user experience by making content more personal and useful. However, these systems must be used carefully to protect privacy and avoid showing the same types of content again and again.



TIDBIT

When YouTube shows you "videos you may like," that's AI learning your interests.



ACTIVITY

"Guess My Choice" Simulation

Each student secretly picks a favorite type of video or item. The teacher shows different choices (e.g., comedy, sports, cooking). Based on previous choices, students try to "recommend" the next item to each other. This mimics how AI makes suggestions based on past behavior.

7.3.5 Autonomous Vehicles (Self-driving Vehicles)

Autonomous vehicles, also called self-driving vehicles, are vehicles that can move on the road without a human driver. These vehicles use AI to understand their surroundings and make decisions on their own. They are equipped with smart sensors, cameras, Global Positioning System (GPS), and radars to detect traffic lights, road signs, other vehicles, and people pedestrian. With the help of these tools and AI, the car can safely speed up, slow down, turn, and stop when needed. They can reduce the number of road accidents caused by human mistakes such as distraction or tiredness. They can also help people who are unable to drive, such as the elderly or disabled.

However, there are also some challenges. These vehicles must be tested many times to make sure they are safe in all situations. There are also concerns about job loss for drivers and the need for strong traffic laws for AI systems. People also worry about who is responsible if a self-driving car causes an accident. That's why engineers, governments, and companies are working together to improve safety and create fair rules for these smart vehicles.



ACTIVITY

"Map the Journey" Planning Task

Students draw a map from their school to a park or market. Then they mark what a self-driving car would need to detect (like traffic lights, people crossing, or road signs).





Figure 7.4: Self-driving car

7.4 Benefits and Challenges of AI Applications

AI applications are helping people complete tasks faster and more easily. In offices, AI systems manage files, answer questions. In education, AI can guide students with smart learning apps that adjust to their level. This saves time and helps people focus on more important decisions. At the same time, AI supports businesses in planning, predicting sales, and giving better service. These smart tools are improving lives by reducing human errors and increasing efficiency in many fields like agriculture, banking, and transport etc.

7.4.1 Limitations and Challenges of AI Applications

AI systems are not always correct and sometimes give wrong results if the low quality or biased. A chatbot, for example, might misunderstand a question if the language is unclear. Fields like law or health, small AI mistakes can cause big problems. That is why careful testing is needed. Another challenge is that AI can replace human workers in many jobs e.g., customer services, data entry and clerical works which may lead to job loss. Also, many AI systems cannot make fair decisions if they are not designed carefully.

DO YOU KNOW?

AI systems can make mistakes if they are trained on incomplete or biased data.

7.5 Ethics, Fairness, and Social Impact of AI

Ethics in AI means using AI in a way that is fair, honest, and safe for everyone. Since AI can make decisions, it is important to make sure those decisions do not hurt people or treat anyone unfairly. For example, if an AI system gives better service to one group of people and ignores others, it is not fair. AI should be developed with care, keeping in mind fairness, safety, and respect for human rights.



TIDBIT

AI can be unfair if not designed properly that's why developers must think about what's right and wrong.

7.5.1 Bias in AI and Algorithms

Bias in AI happens when the system gives unfair results because of the data it was trained on. If the data only represents one group of people, the AI may not understand or treat other groups fairly. For example, a face recognition tool may not work well for all skin colors if it was trained mostly on images of only one type. This shows that data should include different people, languages, and situations. Removing bias is a big step toward making AI more fair and helpful for all.

ACTIVITY

"Same Input, Different Output?" Game
Students are given a pretend AI that gives random or unfair results based on the name, color, or gender of users. They analyze whether this is fair and how real-life AI systems can become biased if not designed carefully.

7.5.2 Fairness and Transparency in AI

Fairness in AI means that the system should treat all users equally, no matter their race, gender, age, or background. This is important in systems that make decisions about input Human Rights school admissions, job interviews, or legal cases. To ensure fairness, AI must be trained on data that includes all related types of people and situations.

Transparency means that people should understand how an AI system works and why it made a certain decision. If an AI rejects a loan application or selects a student, users have the right to know what rules or steps the system followed.

Transparency helps avoid mistakes and gives confidence that the system is working honestly.

ACTIVITY

"Explain Your Choice" AI Role Play
One student plays "AI," and others ask it to make a choice (like selecting a student for a prize). The AI must explain the reason.

7.5.3 Privacy, Data Security, and Surveillance

Privacy means keeping personal information safe and not sharing it without permission. Many AI systems collect data from users, such as names, addresses, location, or search history. If this data is not protected, it can be stolen or misused.

Data security means protecting information from hackers, viruses, or other attacks. AI systems must use strong security methods to keep user data safe. **Surveillance** is when systems or cameras watch people's actions. While it can help in safety, it must be used carefully. Watching people all the time without reason can harm freedom and privacy.

7.5.4 AI and Social Equity (Fairness in Society)

AI should be used in a way that helps all people equally, including the poor, disabled, and those living in remote areas. Sometimes, new technologies like AI are only available to rich or educated groups, while others are left behind. This creates a gap between people. To support fairness in society, AI tools should be designed so everyone can benefit, not just a few.

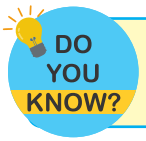
ACTIVITY

"AI for All" Poster Drawing
Students create a poster showing how AI should help everyone ; rich or poor, city or village. Posters should show fairness and equal access.

7.6 Responsibility of AI Designers

AI systems do not make rules by themselves; they follow instructions created by human designers. This means the people who build AI have a big responsibility to make sure their systems are fair, safe, and helpful for society. If an AI system makes a wrong decision or treats someone unfairly, the mistake often comes from the way it was designed. That is why AI designers must think

carefully about how their tools will affect people in real life. Making AI that is helpful and trustworthy is not just a technical job it is a moral duty.



AI is only as good as the people who build it, that's why designers must act responsibly.



"Build a Fair System" Brainstorm
In small groups, students design a fair AI system (e.g., for school admissions or job selection). They list what rules they will follow to avoid bias, mistakes, or harm.

7.6.1 Following Ethics in AI

Ethics in AI means doing what is right and fair when creating or using AI systems. AI designers must make sure their systems do not hurt people, spread false information or treat anyone unfairly. Ethics help designers make decisions that protect users and support justice. Following ethics also includes being honest about how the AI system works. Users should know what data is being used and how decisions are made. Ethical design builds trust between people and technology.



"Right or Wrong?" Ethics Quiz
The teacher reads short AI situations (e.g., spying, unfair selection). Students vote: Is this right or wrong? Then they explain their answer.

Activity (NLP):
Type a sentence in English into Google Translate or a translation app, and convert it into Urdu. Then change a few words and try again.

Activity (Robotics):
Watch a short video of a robot performing a task (like assembling parts, serving food, or cleaning).
After watching, write 3 lines about:

- What the robot did
- How AI helped it
- What would happen if it made a mistake

Activity (Speech Recognition):
Use voice typing on a phone or computer (e.g., Google Docs voice typing). Speak a sentence and watch the text appear. Then speak a different sentence and check if the text is correct.

Summary

- **AI** Field of computer science that teaches machines to "think" and "act" like humans
- **AI capabilities**: Solve problems, learn from data, make smart decisions without step-by-step instructions
- **Real-world applications of AI** : Mobile apps, online shopping, healthcare, transportation, voice assistants
- **Impact on jobs**: Replaces some jobs (factory work) but creates new ones (AI developers, data scientists)
- **Autonomous Vehicles:** Self-driving cars using sensors, cameras, GPS, and AI**.** It Reduces accidents, and provide assistance for disabled/elderly people.
- **Fairness:** Equal treatment regardless of race, gender, age, background
- **Transparency:** Understandable decision-making processes

EXERCISE

Multiple Choice Questions

1. What does AI stand for?
   A) Automatic Intelligence B) Artificial Intelligence
   C) Active Interface D) Applied Information
2. Which of the following is an example of Natural Language Processing (NLP)?
   A) Online calculator B) Voice typing
   C) File copying D) Image editing
3. What type of data helps AI make better decisions?
   A) Short and weak data B) Only images
   C) Large and good quality data D) Data without labels
4. Which tool uses AI to understand human voice?
   A) Video player B) Translator
   C) Speech recognition D) Mouse
5. What is the role of robots in factories?
   A) Watching videos B) Making websites
   C) Doing repeated or dangerous tasks D) Writing stories
6. What is a challenge of using AI?
   A) It works slowly B) It cannot be turned off
   C) It may replace human jobs D) It cannot store data
7. What is bias in AI?
   A) A computer error B) Equal results for everyone
   C) Unfair results due to bad data D) Fast performance
8. Why is fairness important in AI?
   A) So the system looks good B) To avoid slow speed
   C) To give equal treatment to all users D) To use more memory
9. Which of these is NOT an AI tool?
   A) Chatbot B) Face detection
   C) Spell checker D) Paper notebook



Short Questions

1. What is Artificial Intelligence (AI)?
2. How is AI used in daily life?
3. Why does AI need large and good-quality data?
4. What is the role of Natural Language Processing (NLP)?
5. Give one example of robotics in everyday life.
6. What is the purpose of speech recognition technology?
7. How do recommendation systems work?
8. What is meant by bias in AI systems?
9. Why should AI systems be transparent and fair?

Long Questions

1. What is the importance of data in AI? Explain why good and large data is needed.
2. Write in detail about any three real-world applications of AI (such as NLP, robotics, or speech technology).
3. Explain the ethical issues in AI. Why are fairness, privacy, and transparency important?



UNIT 8

Digital Entrepreneurship

Student Learning Outcomes

By the end of this chapter, you will be able to:

- Explain the significance of legal compliance and ethical practices in digital entrepreneurship.
- Identify and describe key data privacy laws.
- Apply ethical decision-making to resolve conflicts of interest and ensure fairness in business operations.
- Recognize responsible marketing practices and explain the necessity of truthful advertising and obtaining user consent for targeted marketing.
- Use structured problem-solving methods, such as root cause analysis and SWOT analysis, to address business challenges.
- Explain the concept of data-driven marketing, including setting measurable goals, defining Key Performance Indicators (KPIs), using analytics tools, and measuring Return on Investment (ROI).
- Describe the role of Customer Relationship Management (CRM) systems in building long-term customer loyalty through personalized communication and loyalty programs.
- Explain how marketing automation tools can improve efficiency, reduce operational costs, and expand customer reach.

Introduction

This chapter provides an understanding of the legal, ethical, and strategic skills that are important to operate a successful digital business. It explains how entrepreneurs can comply with relevant data privacy laws, such as Pakistan's Data Protection Bill, the General Data Protection Regulation (GDPR), and the Children's Online Privacy Protection Act (COPPA). It further outlines the principles of ethical decision-making, conflict of interest management, and balancing profitability with customer welfare.

The chapter also examines the role of digital ethics in promoting safe and respectful online environments for both businesses and customers. Strategic marketing concepts, including data-driven decision-making, customer

relationship management (CRM), competitive analysis, and marketing automation, are discussed with practical examples. By the end of this chapter, you will have the knowledge and skills to make informed, responsible, and strategic business decisions in the digital age.

8.1 From Basics to Advanced Entrepreneurship

In **Grade 9**, you learned that entrepreneurship is the process of designing, launching, and managing a business to create value. We discussed how it drives economic growth, generates employment opportunities, and introduces innovations that improve everyday life. You were also introduced to fundamental digital tools, such as social media platforms, e-commerce websites, and market research applications, that enable entrepreneurs to connect with customers and manage their operations effectively.

In **Grade 10**, we build upon that foundation to explore advanced skills required in the competitive digital business environment. These include legal compliance, to ensure a business operates within the law; regulatory awareness, to keep pace with changing rules and industry requirements; ethical frameworks, structured problem-solving, to address challenges effectively; and strategic marketing, to strengthen a business's presence and relationship with its customers. These skills matter because today's entrepreneurs are expected to protect their intellectual property, respect customer rights, safeguard data privacy, and maintain the trust of their community.

8.2 Problem-Solving & Adaptability

In the business world, challenges can appear at any time. These may include losing customers, facing new competitors, or changes in technology. Successful entrepreneurs do not give up when faced with difficulties. Instead, they use a clear method to understand the problem, find its main cause, and take effective steps to solve it. They also stay flexible and ready to change when conditions demand it. These skills are known as **problem-solving** and **adaptability**.

8.2.1 Structured Problem-Solving

Structured problem-solving means using a step-by-step approach to identify a problem, understand why it is happening, and decide the best solution. This

avoids guesswork and helps prevent the same issue from happening again. This concept has already been explained in detail in the Grade 9 textbook.

Root Cause Analysis

Every problem has a main reason behind it, called the **root cause**. If we only fix the surface problem, it may return. Root cause analysis helps in finding and addressing the main reason, so the solution is long-lasting.

Example: If an online shop receives many complaints, the owner should investigate whether the cause is late delivery, unclear product descriptions, or poor packaging. Once the root cause is found, it can be corrected.

SWOT Analysis

SWOT is a simple tool that stands for:

- **Strengths**, things the business does well (e.g., high-quality products).
- **Weaknesses**, areas that need improvement (e.g., slow customer service).
- **Opportunities**, chances to grow or improve (e.g., high demand for a new product).
- **Threats**, outside factors that can harm the business (e.g., new competitors).

SWOT analysis helps businesses understand their current situation and plan wisely for the future.

TIDBITS

Every challenge is a hidden lesson. Businesses that stay curious, ask **"why"** often, and act quickly turn problems into opportunities for growth.

8.2.2 Adaptability & Resilience

Adaptability means being willing and able to change when needed. **Resilience** means a business can recover quickly after a setback. These two skills help a business in surviving and growing even in difficult times.

Respond to Change in Market or Technology

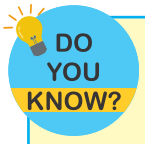
The business environment changes quickly. New customer needs, competitors, or technologies can appear at any time. A business that adapts to these changes has a better chance of success.

Example: A shop that starts home delivery when customers prefer online shopping.

Setbacks Recovery

Mistakes, failures, or losses can happen in any business. A resilient business accepts the setback, learns from it, and works harder to improve.

Example: If a bakery receives complaints about taste, it can improve its recipe, test the new version, and invite customers to try it again.



Some of the world's biggest companies, like **Netflix** and **YouTube**, did not start the way we know them today. Netflix first rented DVDs by mail, and YouTube was made for video dating. They changed their ideas and became famous all over the world.

8.3 Ethical Business Practices & Digital Ethics

Ethics in business means following moral principles and doing what is right, fair, and honest. In the digital age, businesses must ensure that their decisions, marketing, and online activities respect both the law and customer rights. Ethical practices build trust and help a business grow in a responsible way.

8.3.1 Ethical Decision-Making

Ethical decision-making means making choices in business that are fair, honest, and responsible. It is about doing the right thing, even when it may be easier or more profitable to choose another way. A business that follows ethical decision-making earns respect and trust from customers, employees, and the community. Ethical decision-making is not always the easiest path, but it is the one that helps a business grow with integrity, keep loyal customers, and maintain a positive image in society.

Conflicts of Interest

A conflict of interest happens when a person's benefit or relationship can affect their business decisions. This can lead to unfair treatment or bias.

Example: If a shop owner hires their cousin for a job without giving a chance to apply to other skilled people, it's unfair. The fair and ethical way is to let everyone apply, check their skills, and then hire the person who is best for the job.

Balancing Profit with Customer Welfare

A business needs profit to continue operating, but profit must not come from harming customers or risking their safety.

Example: If a business sells food at a high price while using low-quality ingredients, it may earn more money at first. However, customers lose trust, and the business suffers in the future. An ethical business charges a fair price, uses safe and good-quality products, and provides honest information.

Digital Ethics

Digital ethics means using the internet and digital tools responsibly and respectfully. Businesses keep customer information safe, provide accurate details, and interact politely online. Customers also follow rules, treat others with respect, and avoid harmful actions. Businesses maintain safe online spaces by setting clear rules and taking quick action against harassment or bullying.

8.3.2 Responsible Marketing

Responsible marketing means promoting products and services honestly and fairly, providing customers with correct and complete information, and avoiding false promises. This approach helps customers make informed decisions and builds trust in the business.

- Every business advertisement must present truthful and clear information, without exaggerations or hidden conditions that may mislead customers.
- When a business collects personal data of users to show targeted advertisements, businesses must obtain clear permission from customers and explain how the information will be used.

ACTIVITY

Objective:

Students apply ethics by making fair business decisions, protecting customers online, and creating an honest marketing idea.

Steps:

1. **Scenario:**
   "You and your team are running a small online shop for students. Today, you will face three quick challenges. Will your shop stay honest, safe, and trusted?"
2. **Challenges:**
   1. You have two job applicants: a close friend and a more skilled stranger. Decide who to hire and explain why in one sentence.

2. A customer posts rude comments on your shop's social media. Decide how to respond politely and keep your platform safe.
3. Create one short ad line for your business that is truthful, clear, and not misleading.

3. Each group shares its answers. The class votes on the most ethical and trust-building solutions.

8.3.3 Data Privacy Laws

Data privacy laws are rules that protect personal information collected by businesses, such as names, addresses, phone numbers, or payment details.

In Pakistan, the **Personal Data Protection Bill** says that businesses must:

- Collect only the information they need.
- Use the information only for the purpose they explained to the customer.
- Keep the information safe and do not share it without permission.

Some other countries also have strong privacy laws, such as:

- General Data Protection Regulation (**GDPR**) to protect the personal data of people in European countries.
- Children's Online Privacy Protection Act (**COPPA**) to protect the personal data of children under **13**.

These laws matter for Pakistani businesses if they have customers from other countries because they must follow the rules of those countries as well.

Regulatory Awareness

Regulatory awareness means knowing which laws and rules apply to your business and making sure you follow them.

This involves:

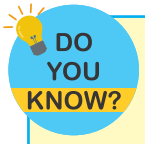
- **Stay Updated:** Checking regularly if any laws have changed that may affect your business.
- **Compliance Audits:** A process where the business is checked to see if it is following all the required laws and rules.

Example: If a school is willing to launch an online portal where students can check their homework, grades, and attendance. To use the portal, students must enter their name, roll number, and contact details.

The school follows data privacy rules by:

- Collecting only the information needed for the portal.

- Using the information only for school purposes, such as sending homework updates or exam schedules.
- Protecting the information with passwords so no one else can access it.



In 2019, **Google** and **YouTube** were fined **$170 million** in the USA for breaking COPPA (Children's Online Privacy Protection Act) rules by collecting personal data from children without their parents' permission. This shows how seriously data privacy laws are enforced worldwide.

8.4 Legal Framework & Compliance

Legal Framework & Compliance means the laws and rules a business must follow to operate legally. It includes registering the business, paying taxes, protecting customer data, and meeting industry requirements. Below are the rules that prevent legal problems and build market trust for any business.

8.4.1 Business Registration in Pakistan

Starting a business in Pakistan is not just about having a good idea. It also needs to be made legal by registering it with the government. Registration makes the business official, allows it to work without legal problems, and helps gain the trust of customers, banks, and other businesses.

Registration with SECP

The Securities and Exchange Commission of Pakistan (SECP) is the main authority to register any new startup business in Pakistan. The registration process consists of the following steps:

- Choose a unique business name and get it approved through the SECP website using the following URL(https://secp.gov.pk/).
- Prepare the required documents, such as the Memorandum and Articles of Association for companies or a partnership deed for partnerships.
- Apply online.
- Get the Certificate of Incorporation, which proves that the business is officially registered.

Obtain National Tax Number

The NTN is an identification number for taxpayers in Pakistan. It is issued by the Federal Board of Revenue (FBR). To obtain an NTN:

- Apply through the FBR's "**Iris"** online portal.
- Provide SECP registration documents, proof of address, and business activity details.
- Once approved, the NTN allows the business to open bank accounts, issue invoices, and meet tax obligations.

Tax Filing and Record-Keeping

Registered businesses must:

- File annual income tax returns and, if applicable, monthly sales tax returns.
- Keep accurate records of sales, purchases, salaries, and expenses.
- Avoid late or incorrect filings, as these can lead to fines or legal actions.

Industry-Specific Regulations

Some businesses in Pakistan must follow special rules in addition to general registration and tax requirements.

Example: Consider two students from your class, Mr. **X** and Mr. **Y** want to start an "online bookstore" for students.

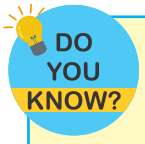
I. SECP Registration: They chose a meaningful and professional name like "E-bookHub" for their business and got it approved on the SECP website. They submit all the required documents online and receive a Certificate of Incorporation.

II. Obtaining NTN: They apply for a National Tax Number (NTN) through the FBR's Iris portal, using their SECP documents, address, and business details.

III. Tax Filing: They keep proper records of sales and expenses and file their annual tax returns on time.

IV. Industry Rules: As an e-commerce business, they add terms of service, a privacy policy, and return/refund rules to their website to follow Pakistan's E-Commerce Policy rules.

By completing these steps, Mr. **X** and Mr. **Y** start their new online bookstore legally, approved by the government, which is more trustworthy for customers and suppliers.



When we register a new business, we need to understand the purpose of above mentioned documents:

I. **Memorandum of Association:** Basic details of a company, like its name, purpose, and location.
II. **Articles of Association:** Rules for running the company.

Partnership Deed: An agreement between two or more people who start a business together, explaining each person's share, role, and responsibilities.

8.4.2 Contracts in Digital Business

A contract is a written or spoken agreement between two or more people or organizations that is legally binding. In digital business, contracts are important because they set out the terms and conditions for working together, whether it is with business partners, suppliers, or customers. They help prevent misunderstandings and provide legal protection if a disagreement occurs.

Basic Elements of a Valid Contract

For a contract to be valid, it must have:

- **Offer**: One side promises to do something, such as sell a product or provide a service.
- **Acceptance:** The other side agrees to the offer without changes.
- **Consideration:** Something of value is exchanged, such as money, goods, or services.
- **Legal Purpose:** The contract must be about something lawful.
- **Consent:** Everyone agrees willingly, without being forced or tricked.

Importance in Digital Business

In digital businesses, contracts can be signed on paper or electronically. They are used for:

- Partnerships, to set clear roles, responsibilities, and how profits will be shared.
- Suppliers, to fix delivery times, payment terms, and quality standards.
- Online Transactions, to state terms of service, refund rules, and customer rights.

Clear contracts build trust and make business dealings smooth and safe.

Example: The students of Class 10th are planning to arrange a farewell party. Mr. **X** and **Y** are volunteers to manage the party arrangements. They decide to make a simple contract, so their responsibilities are clear.

- Offer: Mr. **X** offers to arrange the food and drinks for the party.
- Acceptance: Mr. **Y** agrees to handle decorations and music.
- Consideration: Both will contribute Rs. 1,500 each towards the total cost.
- Legal Purpose: Organizing a class party with the school's permission is lawful.
- Consent: Both agree willingly without any pressure.

Contract Terms:

- Mr. **X** will confirm the food order three days before the event.
- Mr. **Y** completes decorations on the morning of the party.
- Any extra expenses will be shared equally.

ACTIVITY

Objective:

Students will work in groups to plan a small business idea for the "school canteen" and ensure it follows the legal framework and compliance rules. They will also prepare a simple contract to define each member's responsibilities.

Steps:

I. **Form Groups:** Divide the class into groups of 4–5 students.

II. **Business Idea:** Each group will imagine they are starting a small business in the school canteen (e.g., snack counter, juice bar, stationery corner).

III. **Create a Simple Contract:**
Write a basic contract between two group members showing:
- **Offer** ⟶ what each member will do.
- **Acceptance** ⟶ agreement to the plan.
- **Consideration** ⟶ what each member will contribute (e.g., money, time, materials).
- **Legal Purpose** ⟶ lawful activity in school.
- **Consent** ⟶ agreement without pressure

8.4.3 Intellectual Property Rights (IPR)

IPRs are legal rights that protect the creations of a person or business. These creations can include inventions, books, songs, artwork, brand names, or logos.

IPR ensures that no one can use or copy these creations without permission. This protection encourages people to be creative and helps businesses keep their unique identity.

Registration of Intellectual Property in Pakistan

The following fundamentals are required to register an intellectual property in Pakistan:

I. **Trademarks**: Protect brand names, logos, or symbols used to identify goods or services.
   - Apply at the Intellectual Property Organization of Pakistan (IPO Pakistan).
   - Apply with the name or logo, description of goods or services, and the required fee.
   - Once approved, the trademark stops others from using a similar name or logo.

II. **Copyrights**: Protect original creative works such as books, songs, computer programs, or drawings.
   - Apply to the Copyright Office under IPO Pakistan.
   - Send details of the work, proof that you created it, and a copy of the work.
   - Copyright is given automatically when the work is created, but registration gives stronger legal protection.

III. **Patents**: Give legal protection to a new product, design, or method that someone has created. This means only the inventor can make, use, or sell it for a certain time.
   - Apply at the Patent Office under IPO Pakistan.
   - Submit details, drawings, and descriptions of the invention.
   - Once approved, only the inventor can make, use, or sell the invention for a set period.

Protecting Trade Secrets in the Digital Age

A trade secret is private business information that gives an advantage, such as a recipe, a formula, or a list of customers.

- Keep this information secure, both on paper and in digital form.
- Use passwords, encryption, and limit access to authorized people only.

- Ask employees or partners to sign a Non-Disclosure Agreement (NDA) so they cannot share the information.

ACTIVITY

Objective:
To practice the creation of a business logo and prepare a mock trademark application to understand how legal protection stops others from copying a business name or logo.
Steps:

- Design a **unique logo** for their canteen business.
- Prepare a **mock trademark application** that includes:
  - The business name.
  - A short description of the food or services the canteen will provide.
  - A drawing or printed copy of the logo.
- Present the application to the class and explain how having a trademark will protect the business name and logo from being copied by others.

TIDBITS

Think of your business **name** and **logo** as your identity in the market; make them original, easy to recognize, and meaningful. A strong **trademark** not only protects your identity from being copied but also builds customer trust, supports marketing, and gives your business a lasting competitive edge.

8.5 Strategic Marketing & Branding

Strategic marketing is the process of planning and carrying out activities to promote a business in a way that attracts customers and keeps them loyal. In today's digital age, businesses use data, technology, and customer insights to design better marketing strategies, strengthen their brand, and build long-term relationships.

8.5.1 Data-Driven Marketing

Data-driven marketing means making marketing decisions based on facts, numbers, and analysis rather than guesswork. Businesses collect information from customers, websites, and sales records to plan effective campaigns to grow.

- Setting Measurable Goals and KPIs (Key Performance Indicators):

Example: An online clothing store in Lahore sets a target to increase T-shirt sales by 20% within three months. They track progress using KPIs such as the number of T-shirts sold each week, website visits, and customer reviews.

- Using Analytics Tools:
  Example: A Karachi-based bakery uses Google Analytics to see which Facebook ads bring the most customers to their website. They notice that ads showing cakes for birthdays get more clicks than general ads, so they focus more on birthday-themed promotions.
- Measuring ROI (Return on Investment)
  Example: A small bookstore spends Rs. 10,000 on Instagram ads and earns Rs. 25,000 in sales from those ads. By comparing the cost with the profit, they confirm the campaign was successful and decide to repeat the strategy.

8.5.2 Customer Relationship Management (CRM)

CRM is a way of handling all interactions with customers to build trust and create long-term relationships. A good CRM approach helps a business understand its customers better and serve them in a more personal way, as shown in **Figure 8.1**.

- **How CRM Tools Work?**
  Example: A sports shop in Islamabad uses **Zoho CRM** to store customer names, contact details, shoe sizes, and purchase history in one system. When a new sports shoe arrives in the size a customer usually buys, the shop can send them a quick message or email about it.
- **Retaining Customers:**
  Example: A coffee shop in Karachi uses a **loyalty program** where customers get a free drink after every 10 purchases. They also send **personalized emails** with the customer's name and favorite drink offers, making customers feel valued and more likely to return.



Figure 8.1: CRM organizes customer data to build lasting relationships.

DO YOU KNOW?

Zoho CRM is a **cloud-based software** used by over **250,000** businesses in **180+** countries to manage customer relationships, from small startups to large companies, helping them increase sales and improve customer satisfaction.

8.5.3 Marketing Automation

Marketing automation means using special software to handle marketing tasks automatically, without needing to do each step manually. This helps businesses save time, work more efficiently, and stay connected with customers.

Examples:

- **Mailchimp** can automatically send welcome emails to new customers, share special offers, and post updates on social media.
- **HubSpot** can track customer activity, send follow-up messages, and create reports on marketing performance.

Benefits for Small Businesses:

- Saves time by handling routine tasks automatically.
- Reduces costs by lowering the need for extra staff.
- Sends timely updates to customers, improving engagement.

- Helps reach a wider audience with less effort.
- Tracks campaign results to see what works best.

TIDBITS

A **brand** is not just a logo; it is the feeling customers have about your business. Every message, product, and service shapes that feeling.

ACTIVITY

Objective:

Your group has created a new snack to sell in the school tuck shop. You have 20 minutes to plan how to brand and market it.

Steps:

1. **Name and Logo**: Select a catchy product name and draw a simple logo.
2. **Brand Message**: Write one sentence that explains why students should buy it.
3. **Marketing Method:** Choose one way to promote it (poster, slogan, or social media post).
4. **Customer Engagement:** Plan one method to keep customers returning (discounts, free samples, or a quiz).
5. **Outcomes:** Each group will present its plan in 2 minutes. The class will vote for the most appealing product idea..

EXERCISE

Multiple Choice Questions

1. Which is an example of intellectual property?
   A. Shop furniture
   B. Brand logo
   C. Raw materials
   D. Office rent
2. Which of the following laws protects personal data in the European Union?
   A. COPPA
   B. GDPR
   C. DMCA
   D. Cybercrime Act
3. Which document is required to register a trademark?
   A. A bank statement
   B. Proof of product purchase
   C. A copy of the logo and business name
   D. Social media profile link
4. Which of the following is a risk of poor data privacy?
   A. Increased customer trust
   B. Cyberbullying
   C. Identity theft
   D. Lower ad costs
5. Which of the following is an example of personal data?
   A. Office address
   B. Employee ID card number
   C. Store opening hours
   D. Website theme color
6. A conflict of interest occurs when:
   A. Employees share company goals
   B. Personal benefit affects a business decision
   C. A business increases profits
   D. Customers give feedback
7. Which is part of digital ethics?
   A. Using false information in ads
   B. Protecting customer data respectfully
   C. Cyberbullying
   D. Avoid online presence
8. Which of the following is a CRM tool?
   A. Photoshop
   B. Zoh o
   C. Google Maps
   D. Canva
9. Which tool is used for marketing automation?
   A. Mailchimp
   B. PowerPoint
   C. MS Word
   D. Paint
10. In marketing, ROI means:
    A. Return on Investment
    B. Rate of Information
    C. Revenue on Ideas
    D. Reach of Internet

Short Questions

1. How does copyright protect creators in the digital world?
2. Differentiate between a patent and a trademark?
3. Define data privacy.
4. Give one example of ethical decision-making in business.
5. Why is honesty important in business decisions?
6. Explain the purpose of data-driven marketing.
7. What does SWOT analysis help identify?
8. Explain the purpose of setting KPIs in marketing.

Long Questions

1. Explain the process of problem solving and adaptability?
2. Write a note on ethical business practices and digital ethics.
3.
    a) (a) Discuss how CRM looks and loyalty programme help in retaining customers.
    b) (b) Explain Marketing automation and describe their benefit for small businesses.