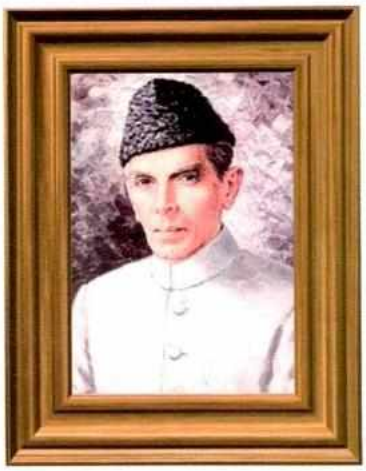
# Computer Science and Entrepreneurship

12




PECTAA
Transformation, Innovation and Excellence

PUNJAB EDUCATION, CURRICULUM, TRAINING AND ASSESSMENT AUTHORITY



"Education is a matter of life and death for Pakistan. The world is progressing so rapidly that without requisite advance in education, not only shall we be left behind others but may be wiped out altogether."

*(September 26, 1947, Karachi)*

Quaid-e-Azam
Muhammad Ali Jinnah رَحْمَةُ اللهِ عَلَيْهِ
Founder of Pakistan






## Instructions for Security Sticker

All textbooks of the Punjab Education, Curriculum, Training and Assessment Authority, carry a rectangular shaped security sticker on the title cover. The sticker exhibits a unique colour shift (orange to green) due to change of angle in the specific circle on it. Moreover, if you scratch protective coating on the specific given space of the sticker with a coin, it will reveal the registered trade name "PECTAA". For further verification, scan the given QR Code with any QR App from your cellphone, then a weblink will be opened to show the original sticker. Look for the security sticker while purchasing the textbooks. If there is no sticker on the title cover or it has been tampered with, do not buy such a textbook and feel free to inform on the given address of PECTAA.

بِسۡمِ اللّٰہِ الرَّحۡمٰنِ الرَّحِیۡمِ

(In the Name of Allah, the Most Merciful, the Most Compassionate.)

# COMPUTER SCIENCE AND ENTREPRENEURSHIP

12



PECTAA
Transformation, Innovation and Excellence

PUNJAB EDUCATION, CURRICULUM, TRAINING AND ASSESSMENT AUTHORITY

Sig1 SideB Process Black 07/07/2026 Computer 12 pecta ishaq

## Authors


- **Dr. Abdul Sattar,**
  Assistant Professor (CS),
  Lahore Garrison University, Lahore.
- **Mr. Zulfiqar Ali Kullachi,**
  Headmaster, Govt. High School,
  Kot Haibat, D.G.Khan


## Editor


- **Prof. Saqib Ubaid,**
  Institute of CS. Khwaja Fareed University of Engineering and IT, Rahim Yar Khan.


## External Review Committee

- **Muhammad Faheem,** Associate Prof. , Govt. Graduate College Gulberg Lahore.
- **M. Zeeshan Shabbir,** Associate Prof. , Govt. Graduate College of Science. Wahdat Road, Lahore.
- **Mahvish Ponum,** Lecturer (CS), Govt. Associate College (W), Gawalmandi, Lahore.
- **Kashif Shahzad Ch,** SST(CS/IT), Govt. Muslim High School No.2, Civil Lines, Lahore.
- **Muhammad Asif Majeed Khan,** SST(IT), Govt. Model High School, Jampur.
- **Mehmood Ahmad,** Computer Teacher, Cresent Model School Lahore.

## Supervision

- **Jahanzaib Khan,** Assistant Director (Compliance Sciences), PECTAA


**Director** (Curriculum & Compliance) — **Aamir Riaz**

EXPERIMENTAL EDITION

**Deputy Director** (Compliance Sciences) — **Syed Saghir-ul-Hassnain Tirmizi**

**Incharge** (Art Cell) — **Mst. Aisha Saidq**

**Layout & Designing** — **Minal Tariq**

**Illustrations** — **Aiyatullah**

**Published By: Tips Academy.**
**Printed By: Ishaq Al-Fateh Printers, Lahore.**

| Date of Printing | Edition | Impression | Copies | Price |
|---|---|---|---|---|
| July 2026 | 1st | 1st | 3500 | 134.00 |

This textbook is based on Revised National Curriculum of Pakistan 2023

**All rights are reserved with Punjab Education, Curriculum, Training and Assessment Authority.**
No part of this textbook can be copied, translated, reproduced or used for preparation of test papers, guidebooks, keynotes and helping books.

# TABLE OF CONTENTS


| Sr No. | Chapters | Page No. |
|---|---|---|
| 1 | Computer Neworks | 1 |
| 2 | Computational Thinking & Algorithms | 20 |
| 3 | Object Oriented Programming Using Python | 33 |
| 4 | Development of Graphical User Interface (GUI) | 45 |
| 5 | Code Testing and Debugging | 60 |
| 6 | Data and Analysis | 72 |
| 7 | Hypothesis Testing | 89 |
| 8 | Applications of Computer Science | 103 |
| 9 | Cybersecurity and Safe Digital Collaboration | 118 |
| ○ | Pairing Scheme | 137 |
| ○ | Model Paper | 139 |

# Chapter 1

# Computer Networks

**Student Learning Outcomes:**

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

- Understand and explain network architecture, its fundamental concepts and become proficient in network diagnostic tools and commands.
- Set up and configure home networks effectively and develop network troubleshooting skills and differentiate between network connection types.
- Understand and measure network bandwidth and latency, and appreciate the importance of network load balancing, network security, data backup and recovery.

## Introduction

This chapter introduces the fundamental concepts of computer networks and explains how devices communicate with each other. Students learn about network architecture, key components, and the basic working of networks. It covers network models and layers, particularly the OSI model, along with different network topologies and types. The chapter also explains common networking devices such as NICs, switches, routers, modems, and wireless access points.

Students are introduced to essential protocols and services like TCP/IP, HTTP, FTP, and DNS, as well as concepts of IP addressing, IPv4/IPv6, subnetting, and NAT. They learn how to set up and configure home networks, including wired and wireless connections, and understand Wi-Fi security and encryption. The use of network diagnostic tools such as ping and ipconfig is also discussed. Finally, the chapter highlights network performance, security measures, and real-world practices to ensure safe and efficient communication.

## 1.1 Introduction to Computer Networks

A computer network allows different devices to connect and communicate with each other. It helps in sharing information quickly and easily. Communication through networks saves time and effort. Computer networks are used in homes, schools, offices, and businesses. Modern life depends heavily on computer networks for daily tasks.

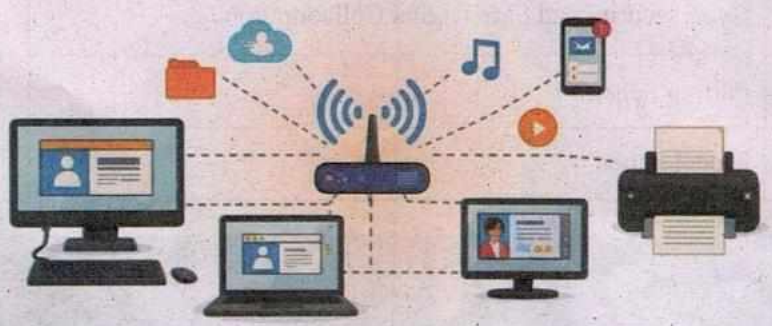


Figure 1.1: A computer network connects multiple devices to share data and resources



Sig1 SideA Process Black 07/17/2026 Computer 12 pectacishaq

Networks can connect computers, mobile phones, printers, and other digital devices (as shown in Figure 1.1). These devices communicate using cables or wireless signals. A network allows data, files, and resources to move from one device to another. Networks can be small or large depending on the need.

**TIDBIT**

Networks support online learning and remote work. Businesses rely on networks for daily operations.

### Uses of Computer Network

Internet browsing is a common example of network use. Mobile phones use networks for calls, messages, and apps. Schools use networks for online classes and learning systems. Banks use networks for online transactions and ATMs. Social media platforms work through computer networks. Smart homes also use networks to control devices.

## 1.2 Network Architecture and Basic Concepts

Network architecture is the overall design of a computer network. It shows how computers, servers, and other devices are arranged (as shown in Figure 1.2). It also defines how data is transmitted across the network. Network architecture includes hardware, software, and communication rules. A good architecture helps the network work efficiently. It also supports future expansion of the network.

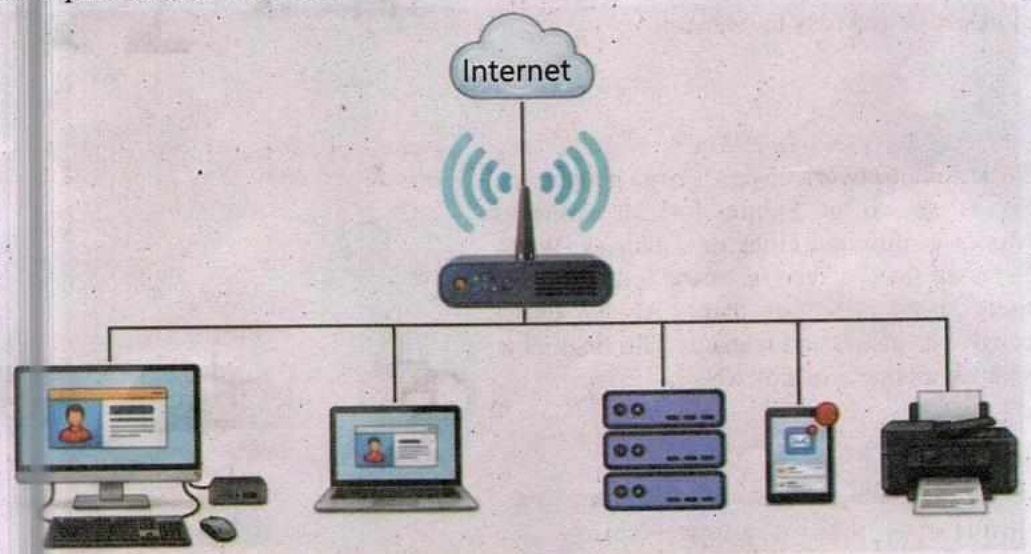



Figure 1.2: Network architecture shows how devices and components are organized in a network

**CLASS ACTIVITY**

Provide pictures or names of devices. Students match device names with functions. Discuss answers

A network works by sending data from one device to another. Data is broken into small packets before transmission. These packets travel through cables or wireless signals. Networking

devices guide packets to the correct destination. The receiving device reassembles the packets into the original data. This process allows smooth and fast communication.

## 1.3 Types of Computer Networks

Computer networks are classified based on size and coverage area. Different types of networks are used for different purposes. Each type connects devices in a specific way. The most common types are LAN, WAN, and the Internet (Public WAN). These networks help in communication and data sharing. Understanding network types helps in proper network selection.

**DO YOU KNOW?**

Computer networks are classified based on distance and coverage area.

### Local Area Network (LAN)

A Local Area Network connects devices within a small area (as shown in Figure 1.3). This area can be a home, school, or office. LAN provides high data transfer speed. It is usually owned and managed by a single organization. Ethernet and Wi-Fi are commonly used in LANs. LAN is cost-effective and easy to maintain.

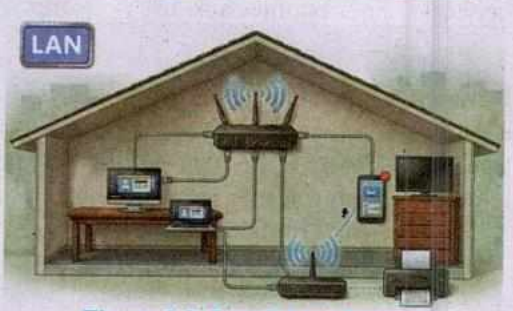



Figure 1.3: Local Area Network

### Wide Area Network (WAN)

A Wide Area Network covers a large geographic area (as shown in Figure 1.4). It connects networks in different cities or countries. WAN uses leased lines, satellites, or the Internet. Data transfer speed is slower than LAN. WAN is expensive to install and manage. The Internet is the largest example of a WAN.

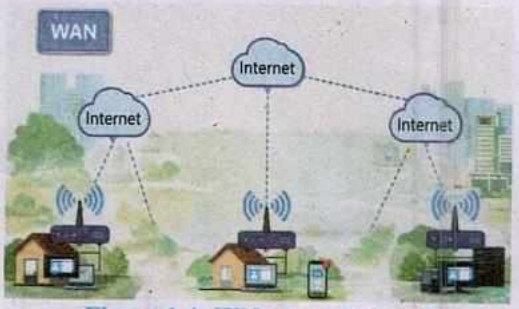



Figure 1.4: Wide Area Network

### The Internet (Public WAN)

The Internet is a global network connecting millions of devices. It allows sharing of information worldwide. Services like email, websites and cloud storage use the Internet (as shown in Figure 1.5). It uses standard protocols for communication. The Internet supports education, business, and entertainment. It plays an important role in daily life.

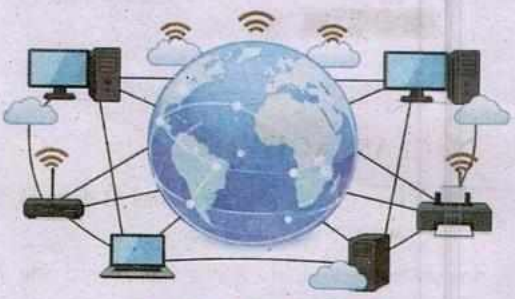


Figure 1.5: Internet

**CLASS ACTIVITY**

Students list services that use the Internet. Discuss its importance.

**TIDBIT**

- DSL(Digital Subscriber Line) : Internet through telephone lines; allows calling and browsing at the same time.
- ISP (Internet Service Provider): Company that provides Internet access to users.

## 1.4 Networking Devices

Networking devices are used to connect computers and other devices in a network. These devices control the flow of data. Each device has a specific role in communication.

### Network Interface Card (NIC)

A Network Interface Card allows a device to connect to a network (as shown in Figure 1.6). It can be wired or wireless. Each NIC has a unique address called MAC address. The NIC sends and receives data from the network. Without a NIC, a device cannot communicate on a network. NICs are built into most modern devices.

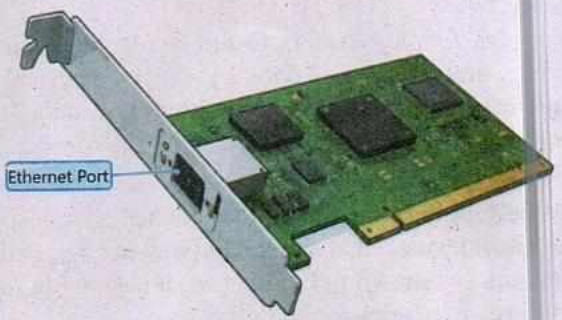



**Figure 1.6: Network Interface Card**

### Switch

A switch connects devices and sends data to the correct device. Switches improve network speed and efficiency. Switches are common in modern networks.

### Router

A router connects different networks together. It directs data packets to their correct destination. Routers connect local networks to the Internet. They use IP addresses to send data. Routers improve network security and control. Home and office networks use routers widely.

### Modem

A modem connects a local network to an Internet Service Provider (ISP). It converts digital data into signals suitable for transmission. It also converts incoming signals back into digital form. Modems are used with cable, DSL, or fiber connections. Internet access depends on a modem. Modems often work with routers.

**CLASS ACTIVITY**

Teacher explains modem using signal conversion example. Students summarize.

### Wireless access point

A wireless access point provides Wi-Fi connectivity. It allows wireless devices to connect to a network (as shown in Figure 1.7). It extends wireless network coverage. Access points are used in large buildings and campuses. They improve signal strength and stability. Wireless access points support mobile connectivity.



Figure 1.7: Wireless Access Point

## 1.5 Network Topologies

Network topology refers to the layout of a computer network. It explains how devices and cables are arranged. Topology can be physical or logical. Physical topology shows actual connections. Logical topology shows how data flows in the network. Topology affects network performance and maintenance.

### Bus Topology

In bus topology, all devices are connected to a single main cable. Data travels along this cable to each devices. It is simple and low in cost. Failure of the main cable can stop the entire network (as shown in Figure 1.8). It is suitable for small networks. Bus topology is not good for large networks.

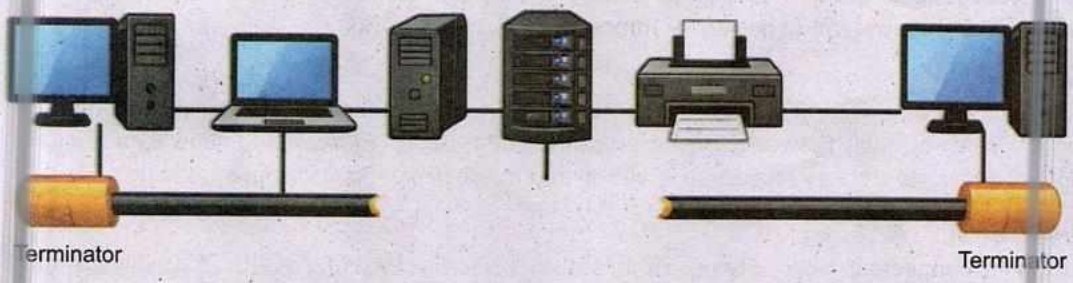



Figure 1.8: Bus Topology

### Star Topology

In star topology, all devices connect to a central hub or switch (as shown in Figure 1.9). Data passes through the central device. It is easy to manage and expand. Failure of one device does not affect others. Failure of the central hub stops the network. Star topology is widely used in modern networks.

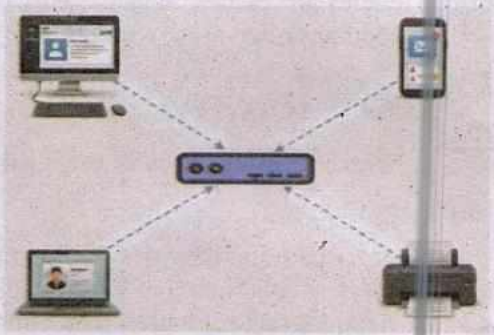


Figure 1.9: Star Topology

## Ring Topology

In ring topology, devices are connected in a circular path (as shown in Figure 1.10). Data moves in one direction around the ring. Each device receives data and passes it along. Failure of one device can affect the whole network. It is more complex than bus topology. Ring topology is rarely used today.

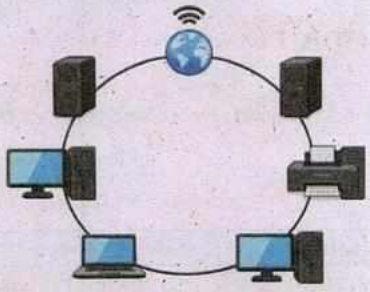


Figure 1.10: Ring Topology

**CLASS ACTIVITY**

Students stand in a circle and pass a message.
Discuss what happens if one student stops.

## Mesh Topology

In mesh topology, devices are connected to many other devices. Multiple paths exist for data transmission. This increases reliability and fault tolerance. Failure of one link does not stop communication. Mesh topology is expensive and complex. It is used in large and critical networks.

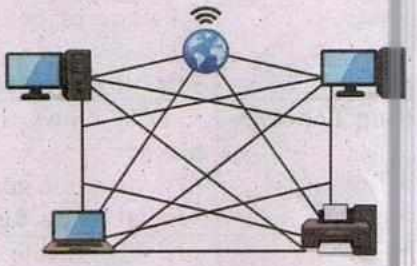


Figure 1.11: Mesh Topology

Figure 1.11 shows a mesh topology network. In this topology, each device is connected to many other devices. Data can travel through multiple paths to reach its destination. If one connection fails, data can use another path. This figure explains why mesh topology is very reliable but complex.

## Tree Topology

In tree topology, devices are arranged in a hierarchical structure like a tree (as shown in Figure 1.12). It combines features of both bus and star topologies, where multiple star networks are connected to a central backbone. Data flows from the root (main node) to branches and sub-branches. It is easy to expand and manage large networks. However, failure of the backbone can affect the entire network. Tree topology is commonly used in large organizations.

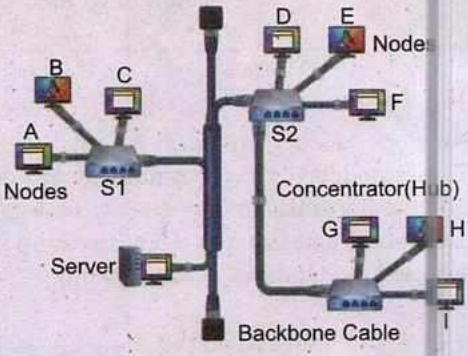



Figure 1.12: Tree Topology

## Advantages and Disadvantages of Network Topologies

Different topologies offer different benefits. Some are easy to install and low-cost (as shown in Table 1.1). Others provide high reliability and performance. Certain topologies are easy to expand. Some are difficult to manage and expensive. Choosing the right topology depends on network needs.

**TIDBIT**

Some topologies are simple and cheap. Others are reliable but costly. Some are easy to expand. Others are difficult to manage. Choice depends on needs.

**Table 1.1: Advantages & Disadvantages of Network Topologies**

| Topology | Advantages | Disadvantages |
|---|---|---|
| **Bus Topology** | • Simple and easy to install<br>• Requires less cable<br>• Cost-effective | • Failure of main cable affects whole network<br>• Difficult to troubleshoot<br>• Performance decreases with more devices |
| **Star Topology** | • Easy to install and manage<br>• Failure of one device does not affect others<br>• Easy to detect faults | • Central device failure stops entire network<br>• Requires more cable<br>• Higher cost |
| **Ring Topology** | • Data flows in an organized manner<br>• No data collision<br>• Equal access for all devices | • Failure of one device affects whole network<br>• Difficult to add/remove devices<br>• Rarely used today |
| **Mesh Topology** | • Highly reliable and fault-tolerant<br>• Multiple paths for data<br>• High security | • Very expensive<br>• Complex installation<br>• Requires large number of cables |
| **Tree Topology** | • Easy to expand<br>• Suitable for large networks<br>• Hierarchical management | • Backbone failure affects whole network<br>• Complex setup<br>• More cabling required |

## 1.6 Network Models and Layers

Network models explain how data moves from one device to another in an organized way. These models divide communication into different layers. Each layer performs a specific task in data transmission (as shown in Figure 1.13). Layered models make networks easier to understand and manage. They also help in troubleshooting network problems. Network models are used as standards in networking.

**DO YOU KNOW?**

Network models are used to understand complex network communication in simple steps.

### Network Layers

Network layers divide the communication process into smaller parts. Each layer has a specific responsibility. Data passes through these layers step by step. This structure makes network

design simple and organized. Changes in one layer do not affect other layers. Layered communication improves reliability and control.

## OSI Model (Open Systems Interconnection)

The OSI model stands for Open Systems Interconnection. It is a standard model used to understand network communication. The OSI model has seven layers. Each layer performs a different function. This model is used for learning and designing networks. It helps different systems communicate properly.

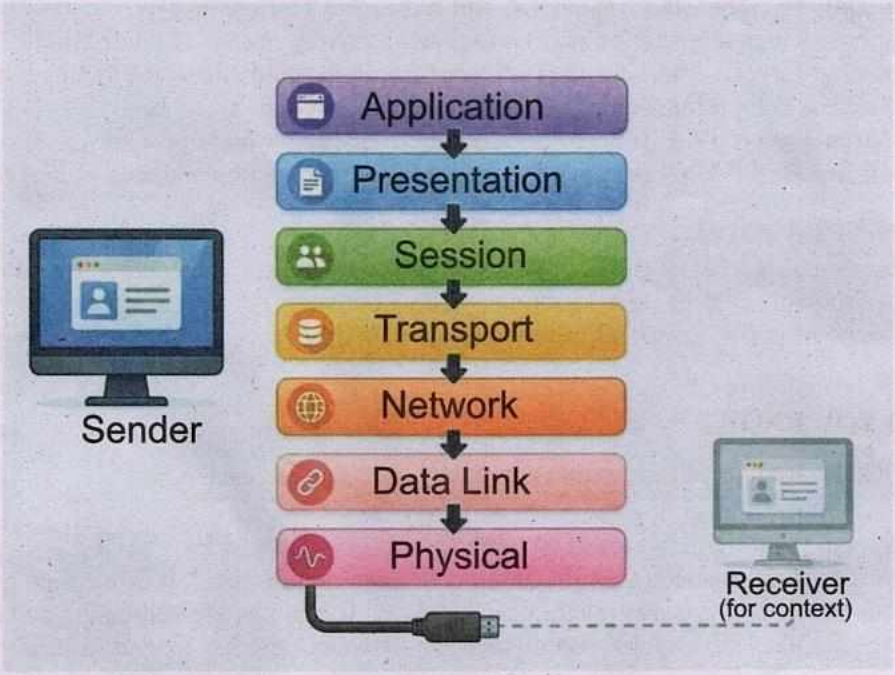



**Figure 1.13: OSI Model on the Sender Side Showing Data Flow from the Application Layer to the Physical Layer**

## Functions of OSI Model Layers

Each OSI layer has a specific role in data communication. The physical layer sends raw data signals. The data link layer controls error-free transfer. The network layer handles routing of data. The transport layer ensures reliable delivery. The session, presentation, and application layers support user interaction and data formatting.

**TIDBIT**

Lower layers handle physical data transfer. Middle layers manage routing and delivery. Upper layers support user applications. Each layer works independently. Together, they ensure proper communication.

Each OSI layer has a specific role in data communication:

1. **Physical Layer:** Sends raw data as electrical, optical, or radio signals over the medium. It defines hardware aspects such as cables, connectors, voltage levels, and data transmission rates.

2. **Data Link Layer:** Ensures error-free data transfer between directly connected nodes. It handles framing, error detection/correction, and controls access to the shared transmission medium.
3. **Network Layer:** Responsible for routing data from source to destination across different networks. It manages logical addressing (IP addresses) and determines the best path for data delivery.
4. **Transport Layer:** Provides reliable and ordered delivery of data between devices. It performs error recovery, flow control, and segmentation of data into smaller units.
5. **Session Layer:** Establishes, maintains, and terminates communication sessions between applications. It manages session checkpoints and recovery in case of interruptions.
6. **Presentation Layer:** Translates data into a format understandable by the application layer. It handles data encryption, compression, and format conversion.
7. **Application Layer:** Provides services directly to end users and applications. It supports protocols like HTTP, FTP, and email services for communication and data exchange.

## 1.7 Network Protocols and Services

A network protocol is a set of rules for communication. It defines how data is formatted and transmitted. Protocols control error handling and data delivery. All devices in a network must follow the same protocol. Protocols ensure smooth data exchange. Examples include TCP/IP and HTTP.

**DO YOU KNOW?**

Protocols are like languages that computers use to talk to each other.

### Transmission Control Protocol and Internet Protocol (TCP/IP)

TCP/IP stands for Transmission Control Protocol and Internet Protocol. It is the main protocol used on the Internet. TCP ensures reliable data delivery. IP handles the addressing and routing of data. Together, TCP/IP enables communication between devices. It is used in almost all networks.

### Hypertext Transfer Protocol (HTTP)

HTTP stands for Hypertext Transfer Protocol. It is used to transfer web pages. HTTP works between a web browser and a web server. It allows users to view websites. HTTP does not encrypt data; secure websites use HTTPS instead. HTTPS is the secure version of HTTP. It uses encryption to protect data transmitted between a web browser and a web server. This encryption is provided using SSL/TLS protocols, which ensure that sensitive information such as passwords, personal details, and payment data cannot be easily intercepted by attackers.

### File Transfer Protocol (FTP)

FTP stands for File Transfer Protocol. It is used to upload and download files. FTP allows file sharing between computers. It is commonly used by web developers. FTP can transfer large files. Secure versions of FTP are also available.

### Domain Name System (DNS)

DNS stands for Domain Name System. It converts website names into IP addresses. DNS makes websites easy to access (as shown in Figure 1.14). Users do not need to remember numbers. DNS servers perform this translation process. It plays a key role in Internet use.

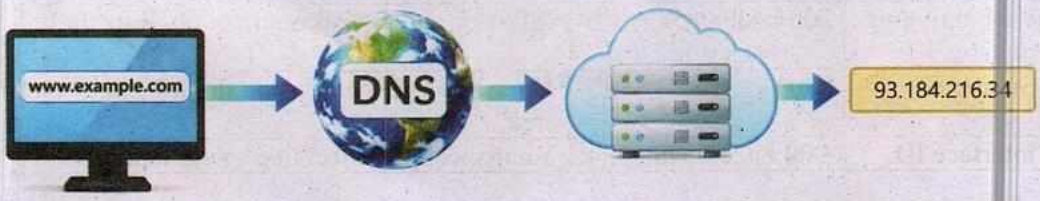



Figure 1.14: DNS Name Resolution Process

**DO YOU KNOW?**

DNS works behind the scenes every time a website is opened.

## 1.8 IP Addressing and Subnetting

An IP address is a unique number assigned to a device. IP stands for Internet Protocol. It helps identify devices on a network. Data uses IP addresses to find its destination. Every network device must have an IP address. Communication is not possible without it.

### IPv4 and IPv6

IPv4 is an older version of IP addressing. It uses 32-bit numeric addresses. IPv4 has a limited number of addresses. IPv6 is the newer version with 128-bit addresses. IPv6 provides more address space. It supports modern Internet needs.

**IPv4**

Example: 192.168.1.10
IPv4 has 4 numbers, each 0–255, separated by dots.
Each number = 1 Byte = 8 bits
Total = 32 bits
Breakdown of IPv4

| Part | Value | What it means |
|---|---|---|
| Network part | 192.168.1 | Identifies your local network, like "Gulshan Block Street 3" |
| Host part | 10 | Identifies your specific device on that network, like "House #10" |

**DO YOU KNOW?**

Total IPv4 addresses possible: 4.29 billion

**IPv6**

Example:
2001:0db8:85a3:0000:0000:8a2e:0370:7334
IPv6 has 8 groups of 4 hexadecimal digits, separated by colons.
Each group = 16 bits
Total = 128 bits
Breakdown of IPv6

| Part | Value | What it means |
|---|---|---|
| Global routing prefix | 2001:0db8:85a3 | ISP/region, like "Pakistan, Sindh, Karachi" |
| Subnet ID | 0000 | Your local network inside the ISP, like "Gulshan area" |
| Interface ID | 0000:8a2e:0370:7334 | Your specific device, like "Your laptop" |

**DO YOU KNOW?**

- IPv6 addresses comprise: $2^{128} \approx 3.4 \times 10^{38}$ total addresses.
- IPv6 will never run out; it is an astronomically massive number.

## Subnetting

Subnetting divides a network into smaller parts. Each part is called a subnet (as shown in Figure 1.15). It reduces network traffic. Subnetting improves performance. It increases security. Large networks use subnetting for management.

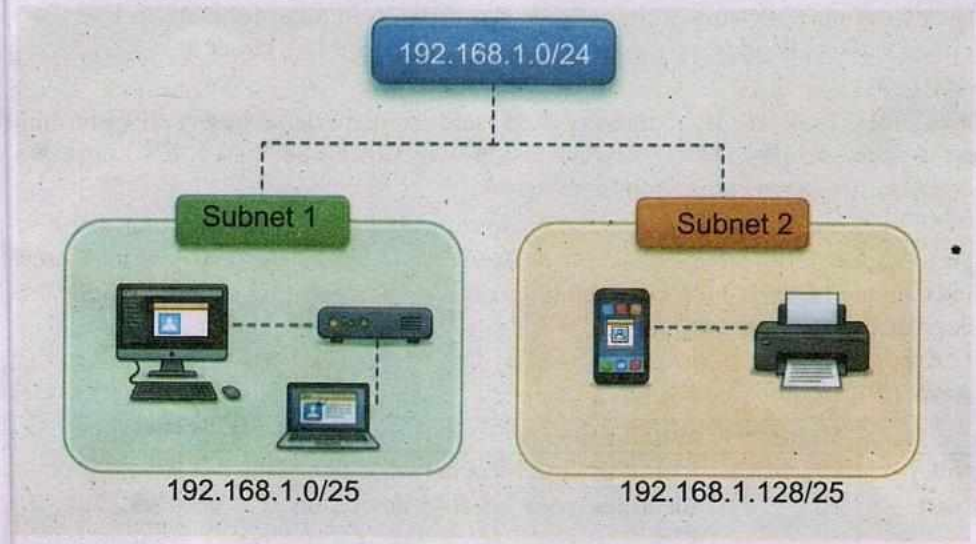



**Figure 1.15: Subnetting Example**

## Network Address Translation (NAT)

NAT stands for Network Address Translation. It allows many devices to share one public IP address. NAT is performed by routers. It hides private IP addresses. This improves network security. NAT is commonly used in networks.

# 1.9 Home Network Setup and Configuration

A home network is used to connect different devices in a house. It helps computers, mobile phones, and printers to communicate with each other. All devices can share the same internet connection. A router is the main device in a home network. Network setup makes data sharing easy and fast. Proper configuration improves network speed and safety.

### Components of a Home Network

A home network has different important components. A router connects all devices in the network. A modem connects the home network to the internet service provider. Computers, laptops, and mobile phones are network devices. Network cables are used for wired connections. Wireless devices use Wi-Fi signals to connect.

### Wired Network Connections

A wired network uses cables to connect devices. Ethernet cables are commonly used in wired networks. Devices are connected directly to a router or switch. Wired connections provide high speed and stable performance. These networks are more secure than wireless networks. However, devices cannot move freely because of cables.

### Wireless Network Connections

A wireless network connects devices without using cables. It uses radio signals to send and receive data. Wi-Fi is the most common wireless technology used in homes. Devices can move freely within the network range. Wireless networks are easy to install and use. They may be slower than wired networks in some cases.

## 1.10 Wi-Fi Security and Router Configuration

Wi-Fi security protects the network from unauthorized access. Without security, anyone can use the internet. Hackers can steal personal information from unsecured networks. Secure Wi-Fi keeps data safe and private. It also prevents slow internet speed. Strong security improves overall network performance.

### Router Web Interface

**1. Router Configuration:** The router web interface is used to manage router settings. It is opened using a web browser. The user enters the router's IP address to access it. A username and password are required to log in. The interface allows changes in Wi-Fi name and password. It also helps control security settings.

**2. Wi-Fi Encryption:** Protects data sent over the network. It converts data into a secure form. WPA2 and WPA3 are common security protocols. Older protocols like WEP are not secure. Strong encryption prevents data theft. Users should always select the latest encryption method.

**3. Default Gateway:** A default gateway connects a local network to other networks. It is usually a router. Data meant for outside networks goes through the gateway. The gateway forwards data to the correct network. Internet access depends on it. Proper gateway setting is necessary.

**CLASS ACTIVITY**

Create sample strong password with length of eight alpha-numeric.

### Guest Networks

A guest network allows visitors to use the internet. It is separate from the main home network. Guest users cannot access personal devices. This increases network safety and privacy. Guest networks help control internet usage. They are useful in homes and offices.

## 1.11 Network Diagnostic Tools and Commands

Network diagnostic tools are used to find network problems. They help check internet and network connections. These tools show whether devices can communicate properly.

Commands are entered using the command prompt or terminal. It helps identify slow or broken connections.

Network troubleshooting is done to find and fix network issues. It helps detect connection and speed problems. Troubleshooting checks cables, devices, and settings. It reduces network downtime and errors. It helps maintain smooth communication. Proper troubleshooting improves network performance.

## Ping Command

The ping command is used to test network connectivity. It checks if a device is reachable on the network. Ping sends small data packets to another device (as shown in Figure 1.16). It measures response time from the destination. This command helps detect network delays. It is commonly used to check internet connection.

**Ping Command:**

C:\> ping google.com

**Sample Output:**

```
Pinging google.com [142.250.190.14] with 32 bytes of data:
Reply from 142.250.190.14: bytes=32 time=25ms TTL=118
Reply from 142.250.190.14: bytes=32 time=24ms TTL=118
Reply from 142.250.190.14: bytes=32 time=26ms TTL=118
Reply from 142.250.190.14: bytes=32 time=25ms TTL=118
```

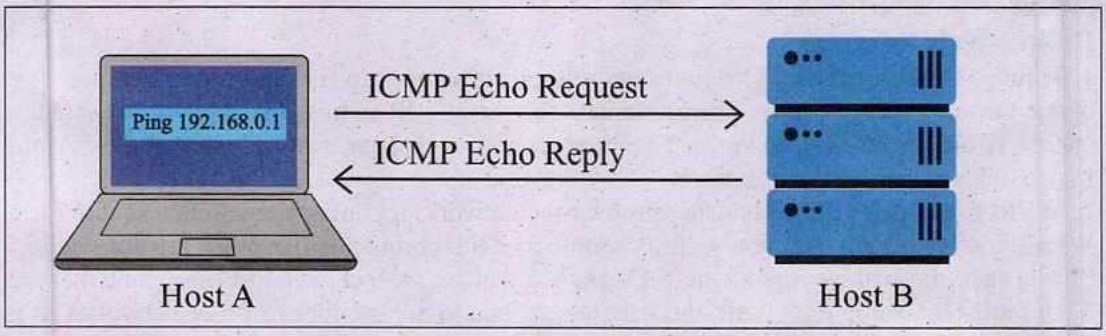



**Figure 1.16: Ping Command**

**CLASS ACTIVITY**

Map packet paths between two locations.

## Ipconfig and ifconfig Commands

Ipconfig is used in Windows operating system. Ifconfig is used in Linux and macOS systems. These commands display network settings. They show IP address and subnet mask. They also show gateway and network status. These commands help in network troubleshooting.

## Telnet, PuTTY, and Secure Shell (SSH)

Telnet is used for remote device access. It sends data without encryption. PuTTY is a popular tool for remote login. SSH is a secure method for remote access. SSH encrypts data for safety. These tools help manage network devices.

## 1.12 Network Performance

Network performance means how well a network works. It shows the speed and quality of data transfer. Good performance allows fast and smooth communication. Poor performance causes slow internet and delays. Network performance depends on many factors, such as bandwidth and latency.

### Network Bandwidth

Network bandwidth is the amount of data sent in a given time. It shows how much data a network can carry. Bandwidth is usually measured in Mbps. Higher bandwidth means faster data transfer. More devices use more bandwidth. Low bandwidth can cause slow internet speed.

### Network Latency

Network latency is the time taken by data to travel. It is the delay between sending and receiving data. Latency is measured in milliseconds. Low latency means faster response. High latency causes lag and delay. Online games and video calls need low latency.

### Causes of Network Delay

Network delay happens due to many reasons. Too many users can slow the network. Poor-quality cables can cause delay. Long distance between devices increases delay. Network congestion affects data speed. Hardware problems also cause network delay.

**CLASS ACTIVITY**

Identify possible causes of delay in a given scenario.

## 1.13 Network Load Balancing

Network load balancing is used to manage network traffic. It distributes work among multiple servers or links. This helps prevent overloading of one device. Load balancing improves network speed and reliability. It is used in large networks and data centers. Proper load balancing ensures smooth performance. It helps networks handle more users.

### Benefits of Load Balancing

Load balancing improves network speed and response time. It reduces the risk of server failure. If one server fails, others continue working. It provides better user experience. Load balancing helps handle high traffic easily. It increases network reliability and availability.

### Basic Load Balancing Methods

There are different methods of load balancing. Round Robin method sends requests one by one. Least Connection method sends traffic to less busy servers. IP Hash method uses IP address to decide traffic. These methods balance the network load. Simple methods are easy to implement.

## 1.14 Network Security

Network security is used to protect computer networks. It keeps data safe from unauthorized access. Security helps prevent data loss and damage. Many users connect to networks daily. Without security, networks become unsafe. Network security uses different tools and rules. It

**CLASS ACTIVITY**

Discuss how security protects sensitive data like CT scans of patients in a hospital.

is important for homes and organizations. Network security protects personal and important information. It stops hackers from entering the network. Security keeps data private and safe. It also protects devices from malware. Secure networks work smoothly without interruptions. Every network needs basic security measures.

### Firewalls

A firewall is a security device or software. It controls incoming and outgoing network traffic. Firewalls block harmful or unwanted data (as shown in Figure 1.17). They act as a barrier between safe and unsafe networks. Firewalls protect computers from attacks. They are used in homes and offices.

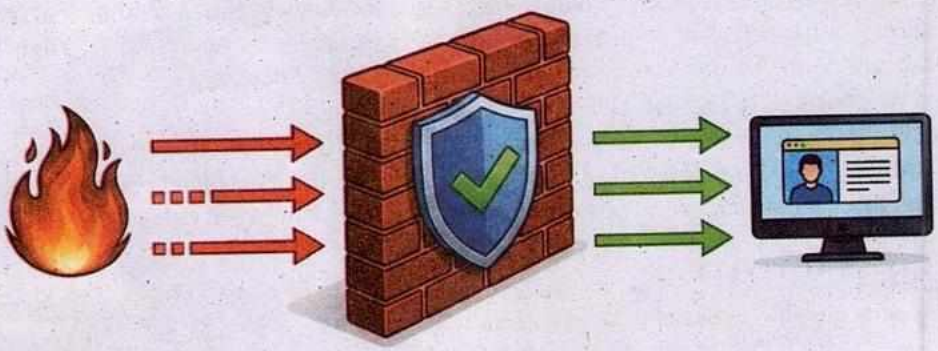


Figure 1.17: Firewall Protection

### Encryption

Encryption changes data into a secret form. Only authorized users can read encrypted data. It protects data during transfer. Encryption is used in emails and online banking. Strong encryption improves data security. It prevents data theft.

### Access Control

Access control decides who can use the network. It allows only authorized users to enter. Usernames and passwords are common access controls. Permissions limit what users can do. Access control protects important resources. It helps manage network usage.

### Common Network Security Threats

There are many threats to network security. Viruses and malware damage systems. Hackers try to steal data. Phishing tricks users to share information. Denial of Service attacks stop network services. Security tools help reduce these threats.

## 1.15 Data Backup and Recovery

Data backup means making a copy of important data. It helps protect data from loss or damage. Data can be lost due to viruses or hardware failure. Backup allows data to be restored easily. Recovery is the process of getting data back. Backup and recovery are very important for safety. Every user should back up data regularly.



Sig2 SideB Process Black 07/17/2026 Computer 12 pecta ishaq

### Types of Data Backup

There are different types of data backup. Full backup saves all data at once. Incremental backup saves only changed data. Differential backup saves changes since last full backup. Backups can be stored on hard drives. Cloud storage is also used for backup.

### Data Recovery Methods

Data recovery is used to restore lost data. Backup files are used for recovery. Recovery software helps get deleted files. System restore can recover system data. Professionals can recover data from damaged devices. Recovery methods help reduce data loss.

## 1.16 Usability and Security Tradeoffs

Usability means how easy a system is to use. A usable system is simple and clear. Users can learn it quickly. Tasks can be done without confusion. Good usability saves time and effort. It improves user satisfaction.

Security means protecting systems and data. It prevents unauthorized access. Security protects against viruses and hackers. It keeps information private and safe. Strong security reduces risks and damage. Every system needs basic security.

### Tradeoffs Between Usability and Security

Increasing security can make systems harder to use. Strong passwords are difficult to remember. Extra login steps slow down users. Simple systems are easier but less secure. Removing security increases risk. A balance is required for safe and easy use.

# Summary

**Computer Network**
A computer network is a group of connected computers and devices that share data, resources, and services.

**Network Architecture**
Network architecture is the design and organisation of a network, including how devices are connected and how communication takes place.

**Network Topology**
Network topology is the arrangement of devices and connections in a network, such as star, bus, ring, mesh, or tree.

**LAN**
A Local Area Network (LAN) connects devices within a small area, such as a home, school, office, or computer lab.

**WAN**
A Wide Area Network (WAN) connects networks over large geographical areas, such as cities, countries, or continents.

**Internet**
The Internet is the largest public network that connects millions of devices and networks worldwide.

**Networking Devices**
Networking devices are hardware components, such as switches, routers, modems, NICs, and access points, used to connect devices and manage communication.

**OSI Model**
The OSI model is a seven-layer model that explains how data travels from one device to another in a network.

**Network Protocol**
A network protocol is a set of rules that controls how data is sent, received, and understood between devices.

**IP Address**
An IP address is a unique address assigned to a device so it can be identified on a network.

**Subnetting**
Subnetting is the process of dividing a large network into smaller networks for better management and security.

**Network Security**
Network security refers to the methods used to protect networks, devices, and data from unauthorized access, misuse, or attacks.

Sig2 SideA Process Black 07/17/2025 Computer 12. pecta ishaq

# EXERCISE

## Multiple Choice Questions

1. A computer network is:
   (a) A single computer system
   (b) A group of connected computers and devices
   (c) A software program
   (d) A storage device

2. The device used to connect a computer to a network is:
   (a) Monitor (b) Keyboard
   (c) Network Interface Card (NIC) (d) Printer

3. A network that covers a small geographical area, such as a school or office, is
   (a) WAN (b) MAN
   (c) LAN (d) Internet

4. The topology that uses a central hub or switch is:
   (a) Bus Topology (b) Ring Topology
   (c) Star Topology (d) Mesh Topology

5. The OSI model contains:
   (a) Five layers (b) Six layers
   (c) Seven layers (d) Eight layers

6. The protocol mainly used to view web pages is:
   (a) FTP (b) DNS
   (c) TCP (d) HTTP

7. The main function of a router is to:
   (a) Store data (b) Connect devices within a network
   (c) Connect different networks (d) Display web pages

8. DNS is used to:
   (a) Transfer files
   (b) Protect data
   (c) Convert domain names into IP addresses
   (d) Connect hardware devices

9. The device that converts digital signals for Internet transmission is:
   (a) Switch (b) Hub
   (c) Modem (d) NIC

10. The topology that provides the highest reliability is:
    (a) Bus topology (b) Ring topology
    (c) Star topology (d) Mesh topology

## Short Questions

1. What is a computer network?
2. Write two uses of computer networks.
3. What is network architecture?
4. Name any two components of a computer network.
5. What is the OSI model?
6. Write the name of any two network topologies.
7. What is a Local Area Network (LAN)?
8. What is the function of a router?
9. What is a network protocol?
10. What is the purpose of DNS?

## Long Questions

1. Define a computer network. Explain its importance and uses in daily life.
2. Explain network architecture and describe the main components of a computer network.
3. What is the OSI model? Explain its layers and their functions.
4. Define network topology. Explain bus, star, ring, and mesh topologies with advantages and disadvantages.
5. Explain the types of computer networks. Describe LAN, WAN, and the Internet.
6. Describe the main networking devices used in a computer network and explain their functions.
7. What are network protocols? Explain TCP/IP, HTTP, FTP, and DNS.
8. Explain IP addressing. Describe IPv4, IPv6, the default gateway, and subnetting.

## Answer Key

1. B
2. C
3. C
4. C
5. C
6. D
7. C
8. C
9. C
10. D

# Chapter 2

# Computational Thinking & Algorithms

**Student Learning Outcomes:**

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

- apply problem-solving by effectively identifying, breaking down, and creating solutions for both straightforward and complex problems using a variety of computational thinking. They will further evaluate and refine their solutions, demonstrating their understanding of different computational techniques.
- understand practical application of fundamental computational artifacts and algorithms. This includes the ability to explain common computational problem-solving methods and a basic comprehension of primitive computational structures like lists, stacks, queues, trees, and graphs. Moreover, students will be able to use knowledge representation and reasoning, employing propositional and predicate logic in their problem-solving.

## Introduction

In this chapter, students learn the basic concepts of logic and computational thinking used in computer science. They understand how to solve problems in a clear and systematic way. Students explore computational thinking techniques such as decomposition and step-by-step problem solving. They also study propositional logic, truth values, truth tables, and logical equivalence.

The chapter introduces satisfiable and unsatisfiable propositions and their role in decision-making. Students then learn predicate logic, variables, and quantifiers to express relationships and conditions. Finally, they develop logical reasoning and inference skills, building a strong foundation for analytical thinking and advanced computer science topics.

## 2.1 Introduction to Computational Thinking

Computational thinking is a way of thinking to solve problems clearly and logically. It helps us break big problems into smaller and simpler parts. This thinking is not only for computer science students.

**TIDBIT**

Computational thinking is used by scientists, engineers, and even chefs when planning complex tasks.

### Computational Thinking

Computational thinking means thinking like a problem solver. It focuses on understanding a problem step by step. We break a big problem into smaller parts. We look for patterns in problems and solutions. We remove unnecessary details and focus on important points. We then create clear steps to solve the problem. These steps can be followed by humans or computers. Computational thinking helps us think logically and clearly.

correct order. This prevents errors during problem solving. Anyone can understand and use the steps easily. Computers also follow step-by-step instructions (as shown in Figure 2.2). Good planning leads to correct results.

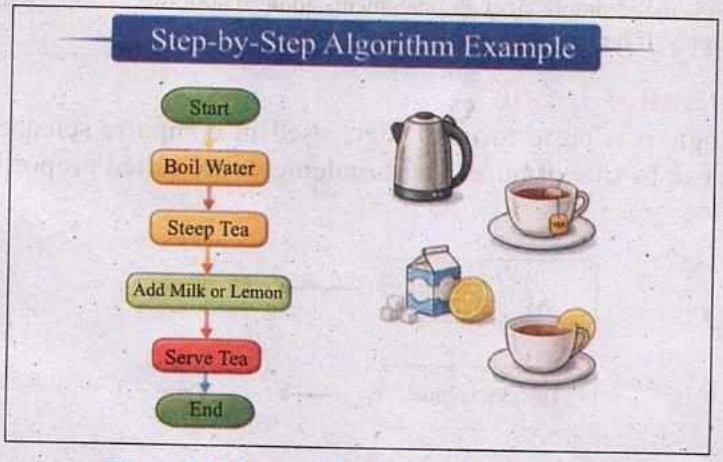



**Figure 2.2: Step by Step Algorithm Example**

### Evaluating and Improving Solutions

After finding a solution, it is important to evaluate it. Evaluation helps us check if the solution is correct. It allows us to find and fix mistakes. Testing the solution shows whether it works properly. We can improve the solution by making it simpler or faster. This process helps us learn from errors. Improving solutions leads to better performance. Evaluation is an important part of problem-solving.

## 2.3 Logic as a Knowledge Representation Framework

Logic is the science of true reasoning and is an important concept in computer science. It helps computers store, understand, and use information correctly. Using logic, complex information can be represented in a simple way.

**TIDBIT**

Logic gives computers a way to "think" about facts and make decisions, almost like a brain.

### Logic in Computer Science

In computer science, logic means using rules to make correct decisions. Logic helps computers understand true and false conditions. It allows systems to process information correctly. Logical statements are used in programs and algorithms. These statements help control the flow of programs. Logic is the basis of many computer operations. It helps computers think in a structured way.

### Role of Logic in Reasoning and Decision Making

Logic plays an important role in reasoning and decision making. It helps computers and humans draw correct conclusions. Using logic, decisions are made based on facts and rules. It removes confusion and guesswork. Logical reasoning helps solve problems step by step. Computers use logic to choose the correct action. Logic ensures that decisions are consistent and accurate. It is used in expert systems and artificial intelligence.

**CLASS ACTIVITY**

- Discuss how logic helps in making decisions.
- Students make simple "if-then" statements about school life.
- Example: "If it rains, then we will stay indoors."

## 2.4 Propositional Logic

Propositional logic is a basic form of logic used in computer science. It deals with statements that can be true or false. These statements are called propositions (as shown in Figure 2.3)

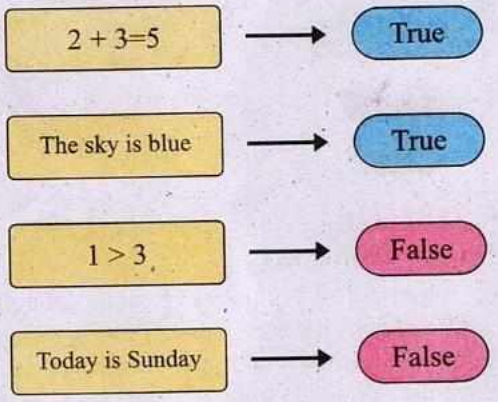



**Figure 2.3: Propositional Logic Statements**

**DO YOU KNOW?**

Propositional logic is the simplest type of logic, but it forms the base of all computer reasoning!

### Proposition

A proposition is a statement that is either true or false. It cannot be both true and false at the same time. Questions and commands are not propositions. For example, "The sun rises in the east" is a proposition. A proposition must have a clear meaning. It should not be confusing. Propositions are the basic units of logic. They are used to form logical expressions. Understanding propositions is important in logic. A proposition is usually represented by a capital letter, such as P, Q, or R. For example, if p represents the statement "The sky is blue", then we write:

P = "The sky is blue"

### Simple and Compound Propositions

A simple proposition contains only one statement. It cannot be broken into smaller parts. For example, "It is raining" is a simple proposition. A compound proposition is formed by combining two or more simple propositions. Logical connectives are used to join them. For example, "It is raining and it is cold" is a compound proposition. Compound propositions can express complex ideas. They are common in logical reasoning. Both types are important in propositional logic.

## Logical connectives: AND, OR, NOT

Logical connectives are words used to connect propositions. AND is used when both statements must be true. OR is used when at least one statement is true. NOT is used to reverse the truth of a statement. These connectives help form compound propositions. They are commonly used in computer programs. Logical connectives help in decision-making (as shown in Figure 2.4). They make logical expressions clear. Understanding them is essential in logic.

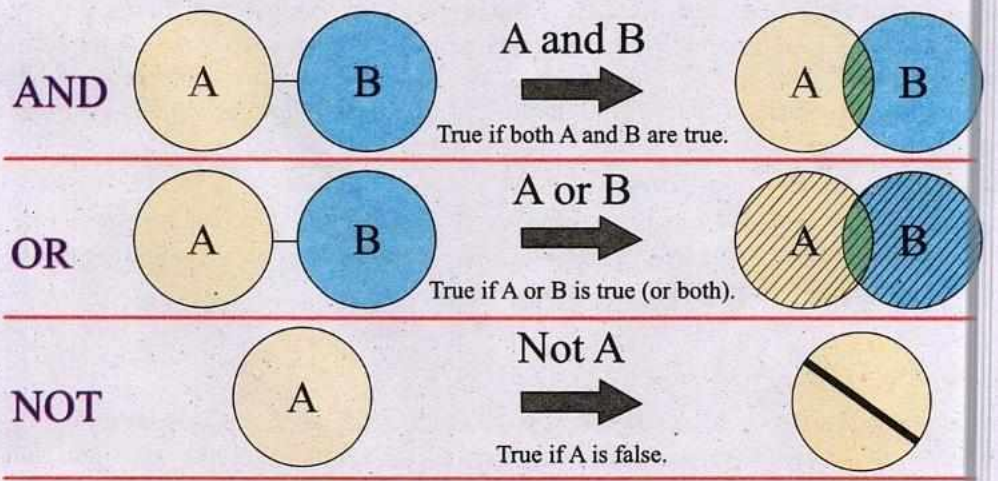



Figure 2.4: Logical Connectives

## 2.5 Truth Tables

Truth tables are used in logic to show the truth values of statements. They help us understand how logical expressions work. A truth table lists all possible combinations of truth values. It shows the final result for each combination. Truth tables are important in propositional logic. They are widely used in computer science and mathematics. They help in checking the correctness of logical expressions. Truth tables make logical thinking clear and simple (as shown in Figure 2.5).

| A | B | A AND B | A OR B | NOT A |
|---|---|---|---|---|
| True | True | True | True | False |
| True | False | False | True | False |
| False | True | False | True | True |
| False | False | False | False | True |

Figure 2.5: Truth Table for Logical Expressions

### Purpose of Truth Tables

The main purpose of truth tables is to show how a logical expression behaves. They help us find when a statement is true or false. Truth tables list all possible input values. They help remove confusion in logical reasoning. Using truth tables, we can check complex expressions easily. They help compare different logical statements. Truth tables are useful in decision-making. They are important tools in logic and computer science.

### Creating Truth Tables for Logical Expressions

Creating a truth table starts with identifying the propositions. Each proposition is given a truth value of true or false. All possible combinations of truth values are listed. Logical connectives are applied step by step. The final result is written for each row. This process helps to understand the expression clearly. Truth tables must be neat and organized. They are easy to follow and use.

### Truth Values of Propositions

Each proposition P or Q can take the value True (T) or False (F). Truth tables enumerate all possible combinations ($2^n$ rows for n propositions). Truth value of a compound proposition is determined by the values of its components.

## 2.6 Propositional Equivalence

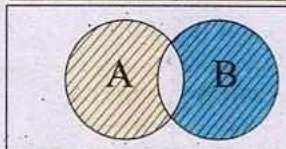
Propositional equivalence is a concept in propositional logic. It explains when two logical expressions have the same meaning. Even if expressions look different, they may give the same result. Propositional equivalence is checked using truth tables. If two expressions have the same truth values in all cases, they are equivalent. The symbol "≡" is used to show equivalence. Propositional equivalence improves clarity in reasoning (as shown in Figure 2.6).

The following two propositional expressions are logically equivalent.

Expression 1: (A AND NOT B) OR (B AND NOT A) ⟺ Expression 2: (A OR B) AND (NOT A OR NOT B)

Both expressions are true in the same cases (XOR).

Venn Diagram Representation




A AND NOT B  B AND NOT A

Neither A nor B  A AND B (not included)

Truth Table

| A | B | (A AND NOT B) OR (B AND NOT A) | (A OR B) AND (NOT A OR NOT B) |
|---|---|---|---|
| F | F | F | F |
| F | T | T | T |
| T | F | T | T |
| T | T | F | F |

Therefore: (A AND NOT B) OR (B AND NOT A) ⟺ (A OR B) AND (NOT A OR NOT B)

**TIDBIT**

Equivalent expressions may look different but always give the same result; they help simplify complex logic.

## Identifying Equivalent Logical Expressions

Equivalent logical expressions are identified using truth tables. Each expression is tested for all possible truth values. If the results match in every case, they are equivalent. Logical laws can also help identify equivalence. These laws simplify expressions. Identifying equivalence helps reduce complex expressions.

## Importance of Equivalence in Logic

Equivalence is important because it helps simplify logic problems. It allows us to replace complex expressions with simpler ones. This saves time and effort. In programming, equivalent expressions improve code efficiency. Equivalence helps avoid logical errors. It ensures correct reasoning and decisions. This concept is useful in mathematics and computer science. Understanding equivalence strengthens logical thinking.

# 2.7 Propositional Satisfiability

Propositional satisfiability is a concept in logic that checks whether a statement can be true. It helps us know if a logical expression has at least one situation where it is true. This concept is widely used in computer science, especially in problem solving and programming. Satisfiability helps us test logical statements and make decisions. It is important for designing algorithms and verifying programs. Understanding satisfiability improves reasoning skills. It also helps in solving complex problems efficiently.

**Example:**

"Bring pizza OR (burger AND do NOT bring soda)."
Result: Bring pizza, no soda = satisfiable expression. Solution exists.
"You must bring pizza AND you must NOT bring pizza."
Result: No solution exists = unsatisfiable expression.

**? DO YOU KNOW?**

A logical expression is satisfiable if there is at least one way to make it true.

## Conditions for a Proposition to be Satisfiable

A proposition is satisfiable if there is at least one combination of truth values that makes it true. If all possible combinations make it false, it is unsatisfiable. Satisfiability depends on the structure of the expression and the logical connectives used. Logical connectives like AND, OR, and NOT determine whether a statement can be true. Checking all possible cases using a truth table is one way to test satisfiability. A proposition with at least one "true" outcome is satisfiable. This condition helps in reasoning and decision-making.

## Examples of Satisfiable and Unsatisfiable Expressions

An example of a satisfiable expression is "P OR Q." This is true if either P or Q is true. Another example is "P AND Q," which is true when both P and Q are true. An unsatisfiable expression is "P AND NOT P." This is never true because P cannot be both true and false. Satisfiable expressions have at least one true outcome, while unsatisfiable expressions have none. Understanding these examples helps in learning logic and problem-solving.

**CLASS ACTIVITY**

Provide several expressions and ask students to classify them as satisfiable or unsatisfiable.

## 2.8 Predicate Logic

Predicate logic is an extension of propositional logic. It allows us to express more detailed statements about objects and their properties. In predicate logic, we can talk about individuals, groups, and relationships between them (as shown in Figure 2.7).

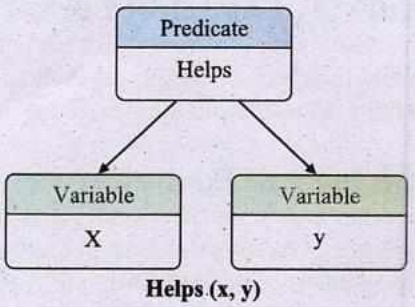



Figure 2.7: Predicate Logic Structure

### Introduction to Predicate Logic

Predicate logic studies statements that contain variables and predicates. A predicate is a statement that can be true or false depending on the values of the variables. For example, "x is greater than 5" is a predicate because it depends on x. Predicate logic allows expressing statements like "All students are hardworking" or "Some books are interesting." It is used to represent knowledge formally.

### Difference Between Propositional and Predicate Logic

Propositional logic deals with simple statements that are either true or false. It cannot describe properties of objects or relationships between them. Predicate logic, on the other hand, uses predicates and variables to express detailed statements. Predicate logic is more expressive and powerful than propositional logic. Propositional logic is easier to understand but less flexible. Predicate logic allows statements about many objects and conditions.

### Predicates and Variables

Predicates describe properties or relationships of objects. For example, "isEven(x)" is true if x is an even number. Variables represent objects or values in the domain of discourse. Using variables, we can write general statements like "For all x, $x > 0$." Predicates and variables together form predicate logic expressions. They help express complex ideas clearly and formally. Understanding predicates and variables is important for reasoning and problem-solving.

**Propositional Logic**

A = "Ali is Tall"

- Cannot see inside (A)
- Limited

**Predicate Logic**

Tall(Ali)

- Can see subject Ali + property Tall
- Much more expressive

## 2.9 Quantifiers in Predicate Logic

Quantifiers are used in predicate logic to describe how many objects satisfy a condition. They help us make general statements or statements about specific objects. Quantifiers make predicate logic more powerful than propositional logic. Using quantifiers, we can write statements like "All students are hardworking" or "Some books are interesting." Quantifiers help represent knowledge clearly and precisely.

**TIDBIT**

Quantifiers like "for all" or "there exists" help us express general rules or possibilities clearly.

### Universal Quantifier (for all)

The universal quantifier is used to indicate that a statement is true for all objects in a given set (as shown in Figure 2.8). It is denoted by the symbol "∀". For example, "∀x, x > 0" means "For all x, x is greater than 0." The universal quantifier is used to make general statements about every object in a domain. It is important in mathematics and logic proofs. Using "for all" ensures that no object in the set violates the statement. This quantifier helps express rules and general truths clearly.

### Existential Quantifier (there exists)

The existential quantifier is used to indicate that a statement is true for at least one object in a set (as shown in Figure 2.8). It is denoted by the symbol "∃". For example, "∃x, x is even" means "There exists at least one x that is even." This quantifier is used when a condition is satisfied by some object but not necessarily all. It helps in expressing possibilities and specific examples. The existential quantifier is important for problem solving and reasoning.

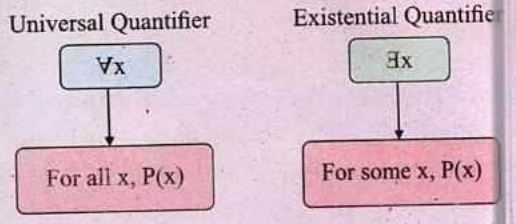



**Figure 2.8: Universal and Existential Quantifier**

### Using Quantifiers in Logical Statements

Quantifiers can be combined with predicates to make clear logical statements. For example, "∀x, P(x)" means the predicate P(x) is true for all x, while "∃x, P(x)" means P(x) is true for at least one x. Quantifiers can also be used together in complex statements. For instance, "∀x ∃y, x < y" means for every x there exists a y that is greater than x. Using quantifiers allows us to express rules, conditions, and relationships clearly.

## 2.10 Applying Predicate Logic to Real-World Problems

Predicate logic can be used to solve real-world problems by representing knowledge clearly and systematically. It helps us model situations involving objects, their properties, and relationships. Using predicate logic, we can analyze problems, make decisions, and find solutions logically.

**? DO YOU KNOW?**

Predicate logic can model family trees, social networks, and even computer programs!

### Representing Relationships Using Predicates

Predicates can represent relationships between objects in the real world. For example, Borrows(x, y) means "student x borrows book y," and Teaches(t, s) means "teacher t teaches student s." These predicates clearly describe how students, books, and teachers are connected.

Using such relationships, we can model the working of the library system. This makes it easier to organize and understand interactions between different entities.

### Writing Logical Statements With Quantifiers

Quantifiers help us write general rules about the library. For example, $\forall x$, Student(x) $\rightarrow$ Borrows(x, Book) means "every student borrows a book." $\exists y$, Book(y) $\wedge$ Available(y) means "there exists at least one book that is available." These statements help define rules and conditions in a precise way. Quantifiers allow us to describe both general situations and specific cases in the library system.

**CLASS ACTIVITY**

Ask students to write rules like "All students must attend class" using universal quantifiers.

### Reasoning and drawing conclusions

Using predicate logic, we can draw conclusions from given facts. For example, if "$\forall x$, Student(x) $\rightarrow$ HasLibraryCard(x)" and "Ali is a student," then we conclude "Ali has a library card."

This logical reasoning helps in decision making, such as checking whether a student can borrow a book. It ensures correct and consistent conclusions, making the system reliable and easy to manage.

## 2.11 Logic-Based Reasoning and Inference

Logic-based reasoning and inference are ways of thinking that help us draw conclusions from known facts. This method uses rules of logic to reach conclusions step by step. Reasoning helps us solve problems, make decisions, and verify information. Logical inference ensures that conclusions are correct and reliable. Understanding this concept improves critical thinking skills. Logic-based reasoning is essential for solving complex real-world problems.

### Logical Inference

Logical inference is the process of deriving new facts from known information. It is a way of reasoning that follows logical rules. For example, if we know "All humans are mortal" and "Socrates is a human," we can infer that "Socrates is mortal." Logical inference ensures conclusions are true if the premises are true. It is used in mathematics, computer science, and artificial intelligence. Inference helps in making decisions based on evidence. It is a fundamental concept in logic and reasoning.

### Deduction Using Logical Rules

Deduction is a type of reasoning that starts from general facts to reach a specific conclusion. Logical rules like "if-then" statements are used in deduction. For example, "If it rains, the ground gets wet" and "It is raining" lead to the conclusion "The ground is wet." Deduction is reliable because the conclusion is guaranteed to be true if the premises are true. Computers use deduction to solve problems and answer questions. Deduction helps in formal reasoning and problem-solving tasks.

# Summary

**Computational Thinking**
Computational thinking is a problem-solving approach in which a problem is understood, broken into smaller parts, and solved step by step using logical thinking.

**Decomposition**
Decomposition is the process of breaking a large or complex problem into smaller and manageable parts.

**Algorithm**
An algorithm is a step-by-step set of instructions used to solve a problem or perform a task.

**Logic**
Logic is a formal method of representing facts, statements, and rules so that correct reasoning and decision-making can be performed.

**Proposition**
A proposition is a statement that is either true or false, but not both at the same time.

**Compound Proposition**
A compound proposition is formed by combining two or more simple propositions using logical operators such as AND, OR, and NOT.

**Truth Table**
A truth table is a table used to show the truth values of a logical expression for all possible combinations of its inputs.

**Propositional Equivalence**
Propositional equivalence means that two logical expressions have the same truth values in all possible cases.

**Propositional Satisfiability**
A proposition is satisfiable if there is at least one condition under which the proposition becomes true.

**Predicate Logic**
Predicate logic is a type of logic that represents statements using subjects, properties, variables, and relationships.

**Predicate**
A predicate is a logical expression that describes a property or relationship of a subject, such as (Tall(Ali)).

**Quantifier**
A quantifier is used in predicate logic to show how many objects a statement applies to, such as for all or there exists.

**Logical Inference**
Logical inference is the process of drawing a correct conclusion from given facts or statements.

**Deduction**
Deduction is a reasoning method in which a specific conclusion is derived from general rules or known facts.

# EXERCISE

## Multiple Choice Questions

1. Computational thinking is mainly used for:
   (a) Playing computer games (b) Solving problems in a logical way
   (c) Writing essays (d) Drawing pictures
2. The first step in problem solving is:
   (a) Writing code (b) Evaluating the solution
   (c) Identifying the problem (d) Breaking the problem
3. Breaking a large problem into smaller parts is called:
   (a) Abstraction (b) Decomposition
   (c) Evaluation (d) Automation
4. A step-by-step solution to a problem is known as an:
   (a) Program (b) Flowchart
   (c) Algorithm (d) Function
5. The logic that uses only true or false values is:
   (a) Predicate logic (b) Fuzzy logic
   (c) Propositional logic (d) Natural logic
6. A compound proposition is:
   (a) (P) (b) (Q)
   (c) $(P \wedge Q)$ (d) True
7. The purpose of a truth table is to:
   (a) Write programs (b) Store data
   (c) Evaluate logical expressions (d) Design algorithms
8. A proposition that is true for at least one case is called:
   (a) Unsatisfiable (b) Invalid
   (c) Satisfiable (d) Equivalent
9. The symbol for the universal quantifier is:
   (a) $\exists$ (b) $\longrightarrow$
   (c) $\wedge$ (d) $\forall$
10. Predicate logic is mainly used to:
    (a) Draw diagrams (b) Represent relationships using variables
    (c) Store numbers (d) Design networks

## Short Questions

1. What is meant by computational thinking?
2. Why is computational thinking important in problem solving?
3. What is decomposition in problem solving?
4. What is an algorithm?
5. What is logic in computer science?

6. What is a proposition?
7. What are truth values?
8. What is a truth table?
9. What is propositional satisfiability?
10. What is a predicate in predicate logic?

## Long Questions

1. Explain computational thinking and discuss its importance in solving problems with suitable examples.
2. Describe the steps involved in problem solving using computational thinking. Explain decomposition with an example.
3. What is logic as a knowledge representation framework? Explain its role in reasoning and decision making.
4. Explain propositional logic. Discuss simple and compound propositions with examples.
5. What are truth tables? Explain how truth tables are created and used to evaluate logical expressions.
6. Explain propositional equivalence. Describe how equivalent logical expressions are identified and why equivalence is important.
7. What is propositional satisfiability? Explain satisfiable and unsatisfiable propositions with examples.
8. Explain predicate logic and quantifiers. Discuss how predicate logic is applied to real-world problems.

## Answer Key

1. B
2. C
3. B
4. C
5. C
6. C
7. C
8. C
9. D
10. B

# Chapter 3

# Object Oriented Programming Using Python

**Student Learning Outcomes:**

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

- Develop Python applications using object-oriented programming (OOP) principles.

## Introduction

In this chapter, students will learn the fundamentals of Object-Oriented Programming (OOP) using Python. They will explore key OOP concepts like classes and objects to model real-world entities in code. The chapter covers encapsulation, which helps protect data and ensure privacy, and inheritance, allowing for code reuse and extension. Students will also study polymorphism to create flexible, adaptable programs. These concepts will help students develop well-structured and maintainable Python applications. By the end of the chapter, students will be proficient in applying OOP principles to design efficient and scalable programs.

## 3.1 Object-Oriented Programming (OOP) Using Python

Object-Oriented Programming is a method of designing programs using objects and classes. Each object represents a real-world entity with specific properties and behaviors. Classes act as templates for creating multiple objects with similar characteristics. OOP emphasizes encapsulation, inheritance, and polymorphism to manage data and behavior efficiently. This approach helps in modeling real-world problems in programming.

**CLASS ACTIVITY**

Create a chart comparing procedural programming and OOP. Have students explain scenarios where OOP is more useful.

### Benefits of OOP: Modularity, Reusability, and Easier Maintenance

OOP allows code to be divided into modules, making it easier to manage. Classes and objects can be reused in multiple programs or parts of a program, reducing repetition. Programs built with OOP are easier to maintain and update because changes in one part do not affect the entire system. The structure of OOP supports scalability and adaptability in software development. It also improves collaboration by making code more understandable and organized.

**TIDBIT**

Object-Oriented Programming helps programmers think in terms of real-life objects, like students or cars, making programs easier to understand and manage.

## 3.2 Core Concepts of OOP

Object-Oriented Programming (OOP) has several core concepts that make programs organized, reusable, and easier to manage. These concepts include classes and objects, encapsulation, inheritance, and polymorphism.



Sig3 SideA Process Black 07/07/2026 Computer 12 pedia fishaq

**DO YOU KNOW?**

A class is like a blueprint, while an object is the real thing created from that blueprint.

## Classes and Objects – Representing Real-World Entities in Code

A class is a blueprint or template that defines the properties (attributes) and actions (methods) of an entity. An object is an instance of a class that holds actual values for its attributes (as shown in Figure 3.1). Classes and objects allow programmers to model real-world entities in code. Using classes and objects helps organize data and related functionality together, making programs more structured.

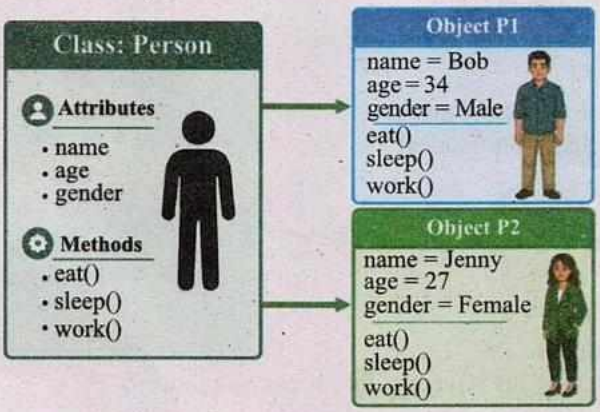



**Figure 3.1: Diagram Showing Class and Object relationship**

**Example**

```
class Student:
    def __init__(self, name):
        self.name = name

    def display(self):
        print("Student Name:", self.name)

# Creating objects
object1 = Student("Ali")
object2 = Student("Ahmed")

# Using objects
object1.display()
object2.display()
```

- class Student → defines a blueprint for students
- __init__ → sets the name when an object is created
- self.name → stores the student's name
- display() → prints the student's name
- object1 and object2 → two different student objects
- Each object has its own data (Ali, Ahmed)

## Encapsulation

Encapsulation is a concept in Object Oriented Programming (OOP) where we bundle data (variables) and methods (functions) into a single unit (class) and restrict direct access to the data.

Unlike languages like Java or C++, Python does not have strict "private" or "protected" keywords. Instead, it relies on naming conventions to signal how data should be accessed.

**TIDBIT**

Hide the data and control access using methods.

### Why Encapsulation is important

- Protects data from unwanted access
- Improves security
- Makes code easier to manage
- Allows controlled modification

**Example:**

Think of a bank account:

Your balance is hidden

You can only access it using:

- deposit()
- withdraw()

You cannot directly change the balance; this is encapsulation.

### 1. Levels of Access Control

In Python, you can control the visibility of class members using specific naming prefixes:

- **Public Members:** These are accessible from anywhere—inside or outside the class. By default, all members in Python are public (as shown in Figure 3.2).
  - *Syntax:* self.attribute_name
- **Protected Members:** These are intended for internal use within the class and its subclasses. It is a convention (not a restriction) to signal to other developers that the member should not be accessed from outside.
  - *Syntax:* self._attribute_name (single underscore)
- **Private Members:** These are highly restricted and cannot be easily accessed from outside the class. Python uses a mechanism called Name Mangling to change the internal name of the variable to prevent accidental access.
  - *Syntax:* self.__attribute_name (double underscore)

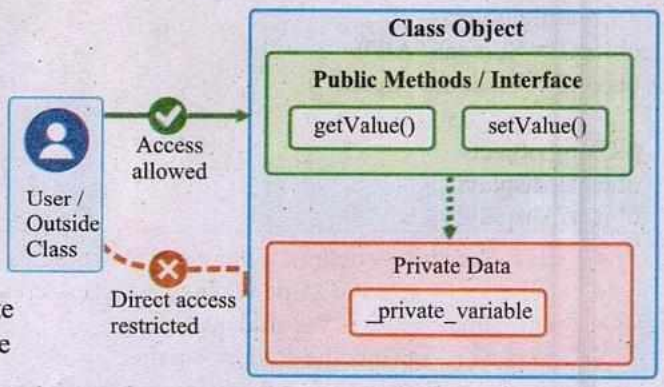



**Figure 3.2: Encapsulation diagram showing private and public variables**

**Example:**

```
class BankAccount:
    def __init__(self, balance):
        self.__balance = balance      # Private variable

    def deposit(self, amount):
        self.__balance += amount
    def withdraw(self, amount):
        if amount <= self.__balance:
            self.__balance -= amount
        else:
            print("Insufficient balance")

    def get_balance(self):
        return self.__balance
# Usage
account = BankAccount(1000)

account.deposit(500)
account.withdraw(300)

print(account.get_balance())          # Output: 1200

# Direct access will fail
# print(account.__balance)            # Error
```

## Inheritance – Reusing and Extending Functionality

Inheritance allows a class to take properties and methods from another class. The new class, called a subclass, can add or modify features of the parent class. Inheritance promotes code reuse and reduces redundancy. It also makes it easier to extend programs without rewriting existing code.
There are the following types of inheritance:

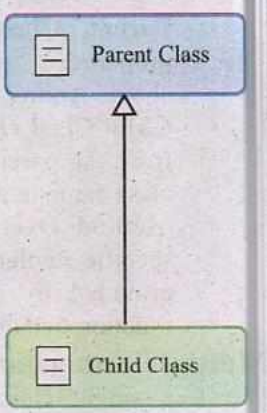


Figure 3.3: Single Inheritance

**Single Inheritance:** Single inheritance occurs when a child class (subclass) inherits attributes and methods from exactly one parent class (superclass). This is the most fundamental form of inheritance in Python, promoting code reusability by allowing the child class to extend or modify the behavior of the parent (as shown in Figure 3.3).

**Example:**

In this example, the Dog class inherits the basic functionality of an Animal but adds its own specific behavior.

```
# Parent class (Superclass)
class Animal:
    def __init__(self, name):
        self.name = name

    def eat(self):
        print(f"{self.name} is eating.")

# Child class (Subclass) inheriting from Animal
class Dog(Animal):
    def bark(self):
        print(f"{self.name} says Woof!")

# Create an object of the child class
my_dog = Dog("Buddy")

# Access method from Parent class
my_dog.eat()  # Output: Buddy is eating.

# Access method from Child class
my_dog.bark() # Output: Buddy says Woof!
```

**TIDBIT**

self is the reference to the current object, passed automatically to instance methods.

- **Parent Class (Base Class):** The class whose properties are inherited. In the example above, Animal is the parent class.
- **Child Class (Derived Class):** The class that inherits from the parent. It is defined by passing the parent class name in parentheses: class Child(Parent):.
- **Method Overriding:** A child class can provide a specific implementation of a method that is already provided by its parent. You can use the super() function to call the parent's version of that method.

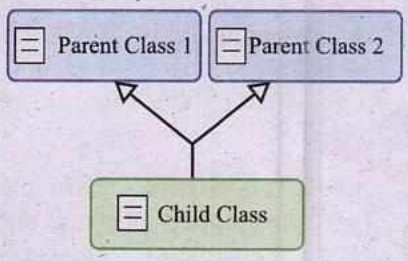



Figure 3.4: Multiple Inheritance

**Multiple Inheritance:** Multiple inheritance in Python allows a single class to inherit features (attributes and methods) from more than one parent class. This enables a child class to combine behaviors from different sources, such as a Smartphone class inheriting from both a Camera and a Computer class (as shown in Figure 3.4).

**Example:**
In this example, the Child class inherits from both Father and Mother classes. It can access methods from both parent classes.

```
lass Father:
  def skill1(self):
      print("Father's skill: Gardening")
lass Mother:
  def skill2(self):
      print("Mother's skill: Cooking")

# Derived class inheriting from two base classes
lass Child(Father, Mother):
  def own_skill(self):
      print("Child's skill: Coding")
# Creating an instance of Child
 = Child()
.skill1()              # Accessing Father's method
.skill2()              # Accessing Mother's method
.own_skill()           # Accessing its own method
```

**Multilevel Inheritance:** Multilevel inheritance in Python occurs when a class inherits from a another drived class, creating a chain of inheritance similar to a family tree (Grandparent → Parent → Child). In this structure, the "grandchild" class has access to all methods and attributes from both its immediate parent and its grandparent (as shown in Figure 3.5).

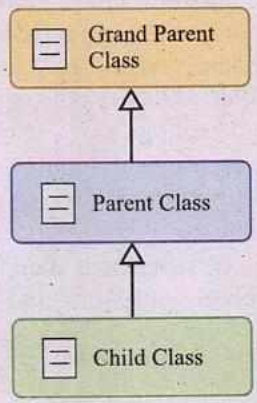



**Figure 3.5: Multilevel Inheritance**

**Example:**
In this example, ElectricCar inherits from Car, which in turn inherits from Vehicle.

```
 Base Class (Grandparent)
lass Vehicle:
  def start_engine(self):
     print("Engine started.")
```

```
# Intermediate Class (Parent)
class Car(Vehicle):
    def drive(self):
        print("Car is driving.")

# Derived Class (Child)
class ElectricCar(Car):
    def charge(self):
        print("Battery is charging.")

# Creating an instance of the bottom-most class
my_tesla = ElectricCar()

# Accessing methods from all levels
my_tesla.start_engine()  # Inherited from Vehicle
my_tesla.drive()         # Inherited from Car
my_tesla.charge()        # Defined in ElectricCar
```

**CLASS ACTIVITY**

Create a Vehicle class and extend it with Car and Bike subclasses. Add unique methods to each subclass and demonstrate inheritance.

## Polymorphism

Polymorphism means "many forms". It refers to the ability of an entity (like a function or object) to perform different actions based on the context. Technically, Polymorphism allows the same method, function, or operator to behave differently depending on the object it is working with. This makes code more flexible and reusable.

**Example:** In a backend payment system, multiple payment options are available, such as Credit Card, UPI, NetBanking, and Wallet. All payment types use a common method named processPayment(), but different implementations:

- NetBanking Payment: redirects to bank login

The method name remains the same, but the action changes based on the payment type.

**Types of Polymorphism**

Polymorphism refers to the ability of the same method or operation to behave differently based on object or context. It mainly includes compile-time and runtime polymorphism (as shown in Figure 3.6).

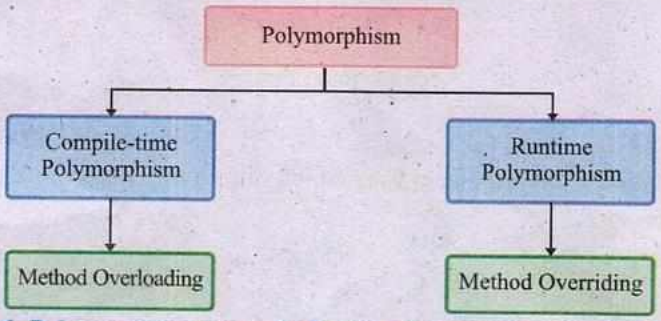



**Figure 3.6: Polymorphism example showing the same function behaving differently**

1. **Compile-time Polymorphism**

Compile-time polymorphism means deciding which method or operation to run during compilation, usually through method or operator overloading. Languages like Java or C++ support this. But python does not support true method overloading based on function signatures like statically typed languages (e.g., Java or C++). Because Python is dynamically typed, a single function can accept arguments of different types, making signature-based overloading unnecessary. Instead, similar behavior can be achieved using default arguments, variable-length arguments (*args, **kwargs), or conditional logic inside the function.

**Example:** This code demonstrates method overloading using default and variable-length arguments. The multiply() method works with different numbers of inputs, mimicking compile-time polymorphism.

```
class Calculator:
    def multiply(self, a=1, b=1, *args):
        result = a * b
        for num in args:
            result *= num
        return result

# Create object
calc = Calculator()

# Using default arguments
print(calc.multiply())
print(calc.multiply(4))

# Using multiple arguments
print(calc.multiply(2, 3))
print(calc.multiply(2, 3, 4))
```

**Output:**

```
1
4
6
24
```

2. **Runtime Polymorphism (Overriding)**

Runtime polymorphism means that the behavior of a method is decided while program is running, based on the object calling it. This happens through Method Overriding a child class provides its own version of a method already defined in the parent class.

**Example:** This code shows runtime polymorphism using method overriding. The sound() method is defined in base class Animal and overridden in Dog and Cat. At runtime, the correct method is called based on object's class.

```
class Animal:
    def sound(self):
        return "Some generic sound"

class Dog(Animal):
    def sound(self):
```

**Output:**

```
Bark
Meow
Some generic sound
```

```
        return "Bark"

class Cat(Animal):
    def sound(self):
        return "Meow"

# Polymorphic behavior
animals = [Dog(), Cat(), Animal()]
for animal in animals:
    print(animal.sound())
```

Here, sound method behaves differently depending on whether object is a Dog, Cat or Animal and this decision happens at runtime. This dynamic nature makes Python particularly powerful for runtime polymorphism.

# Summary

**Object-Oriented Programming**
Object-Oriented Programming (OOP) is a programming approach in which a program is designed using objects that represent real-world entities.
**Class**
A class is a blueprint or template used to create objects. It defines the properties and behaviours of an object.
**Object**
An object is an instance of a class. It represents a real-world entity with its own data and functions.
**Modularity**
Modularity means dividing a program into smaller, manageable parts so that each part can be developed and maintained separately.
**Reusability**
Reusability means using already written code again in different parts of a program or in other programs.
**Encapsulation**
Encapsulation is the process of keeping data and related methods together inside a class. It helps protect data from unauthorized access.
**Inheritance**
Inheritance is an OOP concept in which one class can use the properties and methods of another class.
**Polymorphism**
Polymorphism means using the same method or operation in different ways depending on the object. It helps make code flexible and adaptable.

# EXERCISE

## Multiple Choice Questions

1. The OOP concept that hides data inside a class is:
   (a) Inheritance (b) Polymorphism
   (c) Encapsulation (d) Abstraction
2. Classes and objects in Object-Oriented Programming are best described as:
   (a) Classes are real-world entities, and objects are methods.
   (b) Classes are blueprints, and objects are instances of those blueprints.
   (c) Classes are used for data storage, and objects are used for functionality.
   (d) Objects define functionality, and classes store data.
3. The main idea behind encapsulation in OOP is:
   (a) To hide data inside a class and prevent unauthorised access.
   (b) To create a flexible and adaptable programme.
   (c) To reuse code from other classes.
   (d) To allow multiple inheritance in classes.
4. Inheritance in OOP means:
   (a) Reusing and extending functionality from a parent class to a child class.
   (b) Hiding data in a class.
   (c) Creating multiple instances of the same class.
   (d) Managing polymorphic behaviour of classes.
5. Polymorphism in OOP is best defined as:
   (a) Allowing an object to take multiple forms through method overriding.
   (b) Hiding data inside a class for security.
   (c) Using multiple inheritance to extend classes.
   (d) Ensuring that each object has only one method.
6. An example of inheritance in Python is:
   (a) Creating two classes with similar attributes.
   (b) A child class inheriting attributes and methods from a parent class.
   (c) Using private variables within a class.
   (d) Modifying a method inside the same class.
7. The purpose of polymorphism in programming is:
   (a) To create reusable functions with multiple parameter types.
   (b) To allow a function or method to have different implementations.
   (c) To combine different classes into a single class.
   (d) To manage the flow of control in a programme.
8. The keyword used to define a class in Python is:
   (a) def (b) class
   (c) function (d) object

9. Encapsulation is best described as:
   (a) Combining data and methods into a single unit and controlling access to data.
   (b) Combining multiple classes into one class.
   (c) Inheriting methods from other classes.
   (d) Allowing objects to have multiple forms.
10. Inheritance allows a class to:
   (a) Hide its data only.
   (b) Inherit attributes and methods from another class.
   (c) Define multiple functions only.
   (d) Create objects without a class.

## Short Questions

1. What is Object-Oriented Programming (OOP)?
2. What is the purpose of a class in Python?
3. What is the difference between a class and an object?
4. What is purpose of __init__...?
5. What is encapsulation in OOP, and why is it important?
6. How does encapsulation help in keeping data secure and private?
7. What is inheritance in OOP, and how does it help in reusing functionality?
8. How does inheritance enable the extension of functionality in Python?
9. What is polymorphism in OOP, and how does it make code more flexible?
10. How does polymorphism help in creating adaptable and reusable code?

## Long Questions

1. Explain the concept of classes and objects in Object-Oriented Programming (OOP).
2. What is encapsulation in OOP?
3. How does inheritance work in OOP?
4. What are the advantages of using inheritance in Python?
5. What is polymorphism in OOP, and how does it enhance flexibility in code?
6. How can encapsulation help improve data security in OOP?
7. How does the combination of inheritance, encapsulation, and polymorphism contribute to better software design?
8. What are the key differences between a class and an object in Object-Oriented Programming?

## Answer Key

| | |
|---|---|
| 1. C | 6. B |
| 2. B | 7. B |
| 3. A | 8. B |
| 4. A | 9. A |
| 5. A | 10. B |

# Chapter 4

# Development of Graphical User Interface (GUI)

**Student Learning Outcomes:**

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

- Design interactive GUI-based programs using the Tkinter library.
- Connect Python applications to databases and perform CRUD operations.

## Introduction

In this chapter, students will learn to develop user-friendly Graphical User Interfaces (GUIs) using Tkinter, Python's built-in GUI toolkit. They will create windows, frames, labels, buttons, entry fields, and manage layouts using pack(), grid(), and place(). They will practice event handling by linking user actions with functions and developing a simple login form. The chapter also introduces database concepts, SQL, Python database connectivity, and basic CRUD operations using SQLite or MySQL.

## 4.1 Graphical User Interface (GUI) Development with Tkinter

Graphical User Interface (GUI) development allows programs to interact with users through visual elements like windows, buttons, labels, and text boxes (as shown in Figure 4.1). GUIs make applications more user-friendly and easier to use compared to command-line interfaces. Python provides tools to create GUI applications efficiently. Tkinter is Python's built-in library for creating GUI programs. It is simple, widely used, and allows developers to design interactive applications quickly.

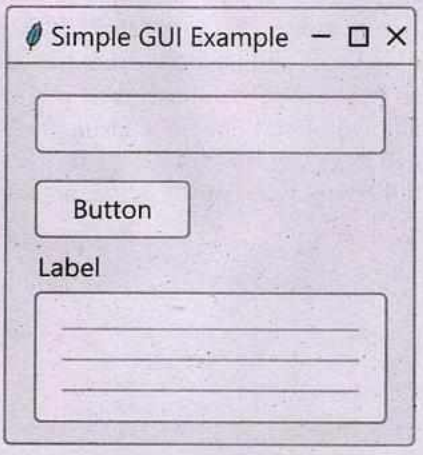



Figure 4.1: Graphical User Interface (GUI)

## What is a GUI? Why is it important?

A GUI, or Graphical User Interface, is a visual interface that allows users to interact with software using elements like buttons, menus, and forms. GUIs are important because they make programs easier to understand and operate, especially for non-technical users. They improve usability by providing visual feedback and reducing the need to remember complex commands. GUI applications are widely used in desktop software, web applications, and mobile apps.

## Overview of Tkinter as Python's built-in GUI toolkit

Tkinter is Python's standard GUI toolkit that comes pre-installed with the language. It provides a set of classes and functions to create windows, buttons, labels, entry fields, and other widgets. Tkinter is easy to learn, lightweight, and supports event-driven programming, where actions are triggered by user events. It is suitable for building small to medium-sized GUI applications quickly and efficiently.

**Example:**

**Code:**

```
import tkinter as tk

# Create main window
window = tk.Tk()
window.title("Simple GUI Example")
window.geometry("300x200")

# Function to handle button click (event)
def on_click():
    label.config(text="Button Clicked!")

# Create a label
label = tk.Label(window, text="Welcome
to Tkinter")
label.pack(pady=10)

# Create an entry field
entry = tk.Entry(window)
entry.pack(pady=10)

# Create a button
button = tk.Button(window, text="Click
Me", command=on_click)
button.pack(pady=10)

# Run the application
window.mainloop()
```

**Output:**




- Tk() → creates main window
- Label → displays text
- Entry → takes user input
- Button → performs action

- command=on_click → event handling (button click)
- mainloop() → runs the GUI

## Tkinter Components and Widgets

Tkinter provides many components, called widgets, that help build interactive graphical applications. A program can contain multiple visual elements such as windows, frames, buttons, and text fields. These widgets allow users to enter data, choose options, or perform actions. Each widget has its own purpose, and combining them creates a complete interface. Tkinter also supports layout management, so widgets can be arranged neatly on the screen.

**CLASS ACTIVITY**

Create a main window and divide it into two frames. Add different widgets in each frame to organize layout.

### Creating Windows and Adding Frames

A Tkinter program begins by creating a main window, which serves as the base for all other elements. The window can be resized, given a title, and designed to hold different widgets. Frames are sections within the window that help organize content. Frames divide the window into smaller parts, making it easier to group related widgets. This structure improves clarity and keeps the interface well-arranged.

**Example:**

**Code:**

```
import tkinter as tk

# Create main window
window = tk.Tk()
window.title("Tkinter Frames Example")
window.geometry("400x300")

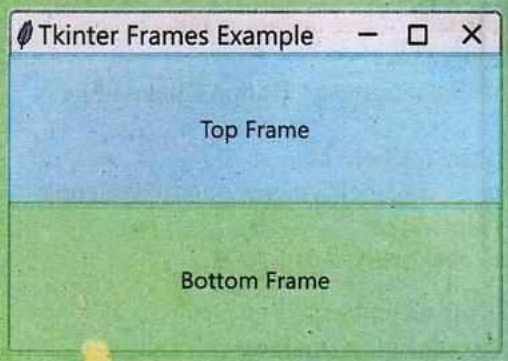
# Create top frame
top_frame = tk.Frame(window, bg="lightblue", height=100)
top_frame.pack(fill="x")

# Create bottom frame
bottom_frame = tk.Frame(window, bg="lightgreen", height=200)
bottom_frame.pack(fill="both", expand=True)

# Add widgets to top frame
label_top = tk.Label(top_frame, text="Top Frame", bg="lightblue")
label_top.pack(pady=20)

# Add widgets to bottom frame
```

**Output:**



```
label_bottom = tk.Label(bottom_frame,
text="Bottom Frame", bg="lightgreen")
label_bottom.pack(pady=50)

# Run the application
window.mainloop()
```

- Tk() → creates main window
- Frame → divides window into sections
- top_frame and bottom_frame → organize layout
- pack() → places frames and widgets
- Frames help keep the GUI clean and well-structured

### Common Widgets: Labels, Buttons, Entry Fields, Menus, and List boxes

Widgets are the basic elements used to build a graphical user interface. A Label is used to display text or messages on the screen. A Button allows the user to perform an action when it is clicked. An Entry field is used to enter text, such as a name or password. A Menu provides a list of options at the top of the window. A Listbox shows a list of items and allows selection from the list.

## Layout Management

Layout management is used to arrange widgets inside a window in a proper way. It decides the position and size of each widget on the screen. Good layout management makes the interface clear and easy to use. It also helps in keeping the window organized. Tkinter provides different methods to manage layout. Each method has its own way of placing widgets.

**TIDBIT**

Proper layout management helps make applications look neat and work well on different screen sizes

## Organizing Elements using pack(), grid(), and place()

The pack() method places widgets in a simple vertical or horizontal order. It is easy to use and works well for small applications. The grid() method arranges widgets in rows and columns like a table. It is useful for forms and structured layouts. The place() method positions widgets at exact locations on the window (as shown in Figure 4.2). It gives more control but needs careful adjustment.




**Figure 4.2: Diagram showing Tkinter Layout Managers: pack(), grid(), place()**

**Example pack():**

**Code:**

```
import tkinter as tk

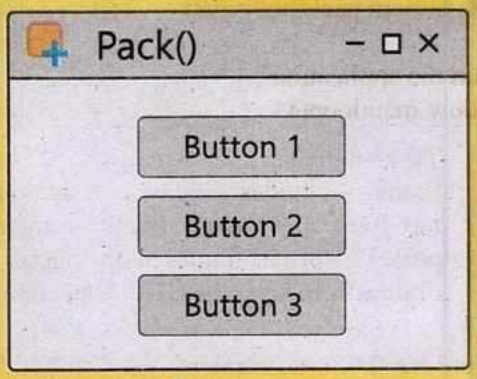
# Create main window
window = tk.Tk()
window.title("pack()")
window.geometry("300x220")

# Create buttons
button1 = tk.Button(window,
text="Button 1", width=12)
button2 = tk.Button(window,
text="Button 2", width=12)
button3 = tk.Button(window,
text="Button 3", width=12)

# Arrange buttons using pack()
button1.pack(pady=10)
button2.pack(pady=10)
button3.pack(pady=10)

# Run the application
window.mainloop()
```

**Output:**




**Example grid():**

**Code:**

```
import tkinter as tk

# Create main window
window = tk.Tk()
window.title("grid()")
window.geometry("300x220")

# Create buttons
button1 = tk.Button(window,
text="Button 1", width=10)
button2 = tk.Button(window,
text="Button 2", width=10)
button3 = tk.Button(window,
text="Button 3", width=24)

# Arrange buttons using grid()
button1.grid(row=0, column=0, padx=10,
pady=20)
```

**Output:**




Sig4 SideA Process Black 07/07/2026 Computer 12 pedia ishaq

```
button2.grid(row=0, column=1, padx=10,
pady=20)

# Button 3 spans two columns
button3.grid(row=1, column=0,
columnspan=2, padx=10, pady=10)

# Keep content centred
window.grid_columnconfigure(0,
weight=1)
window.grid_columnconfigure(1,
weight=1)

# Run the application
window.mainloop()
```

**Example place():**

**Code:**

```
import tkinter as tk

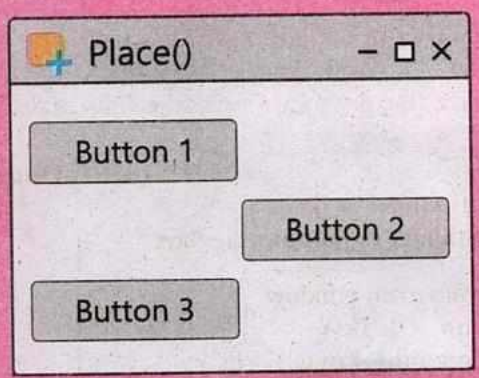
# Create main window
window = tk.Tk()
window.title("place()")
window.geometry("300x220")

# Create buttons
button1 = tk.Button(window,
text="Button 1", width=10)
button2 = tk.Button(window,
text="Button 2", width=10)
button3 = tk.Button(window,
text="Button 3", width=10)

# Arrange buttons using place()
button1.place(x=30, y=40)
button2.place(x=170, y=85)
button3.place(x=30, y=140)

# Run the application
window.mainloop()
```

**Output:**




## Designing Clean and Responsive Interfaces

A clean interface has well-arranged widgets and enough space between them. Text and buttons should be clear and easy to read. Responsive interfaces adjust their layout when the window size changes. This helps the program work well on different screen sizes. Proper alignment improves the overall look of the application. Simple design makes the interface user-friendly.

## Event Handling and Interactivity

Event handling and interactivity allow a program to respond to user actions. An event is an action, such as clicking a button or typing text. The program waits for an event instead of running all the time. When an event occurs, a specific task is performed. This makes the application active and useful. Event handling is important in graphical programs. It helps control how the program behaves.

## Understanding Event-Driven Programming

Event-driven programming is based on actions called events. The program remains idle until an event occurs. Common events include button clicks and key presses. Each event is linked to a specific function. The function runs only when the event happens. This approach saves system resources. It is widely used in graphical user interfaces.

## Handling User Input and Connecting Widgets to Functions

User input is given through widgets like entry fields and buttons. Widgets are connected to functions in the program. When the widget is used, the connected function executes. This allows the program to process the input. For example, a button click can display a message. Proper connections ensure correct program behavior. This interaction improves user experience.

### Example: Simple Login Form

A simple login form allows a user to enter a username and password. Entry fields are used to take the input from the user. A button is provided to submit the information. When the button is clicked, the program checks the entered data. A message is then shown for success or error.

**Example:**

**Code:**

```
import tkinter as tk
from tkinter import messagebox

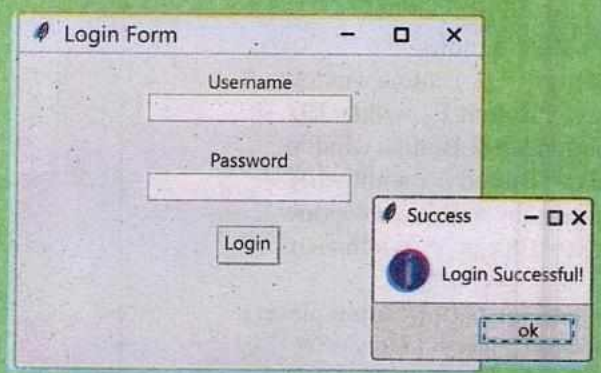
# Create main window
window = tk.Tk()
window.title("Login Form")
window.geometry("300x200")

# Function to check login
def check_login():
    username = entry_user.get()
    password = entry_pass.get()

    # simple check
    if username == "admin" and
password == "1234":

messagebox.showinfo("Success",
"Login Successful!")
    else:
        messagebox.showerror("Error",
"Invalid Username or Password")
```

**Output:**



```
# Username label and entry
tk.Label(window,
text="Username").pack()
entry_user = tk.Entry(window)
entry_user.pack()

# Password label and entry
tk.Label(window,
text="Password").pack()
entry_pass = tk.Entry(window,
show="*")
entry_pass.pack()

# Login button
tk.Button(window, text="Login",
command=check_login).pack(pady=
10)

# Run application
window.mainloop()
```

- Entry → takes username and password
- Button → submits data
- check_login() → checks input
- messagebox → shows success or error
- Works on event-driven programming

**CLASS ACTIVITY**

Explain event-driven concepts and have students add a button that triggers a label update on click.

## 4.2 Working with Databases in Python

Working with databases in Python involves storing and managing data using database systems. Python programs can connect to databases to insert, retrieve, update, and delete data. Databases allow applications to handle large amounts of information efficiently. Using databases helps maintain data accuracy and consistency. Python supports database operations through built-in and external libraries.

**DO YOU KNOW?**

Databases allow programs to store large amounts of information safely and retrieve it quickly when needed.

### Introduction to Databases

A database is an organized collection of data stored in a structured format. It allows data to be accessed and managed easily. Databases reduce data duplication and improve reliability. They

support secure storage of information. Databases are commonly used in many types of software applications.

**Entity:** An entity is a real-world object, person, place, event, or concept about which data is stored in a database.

Example: Student, Teacher, Course, Book, Employee

**Attribute:** An attribute is a property or characteristic of an entity. It describes specific information about an entity.

Example: For the entity Student, attributes can be:

- Student Name
- Roll Number
- Class
- Age

**Relationship:** A relationship shows how two or more entities are connected with each other.

Example: A Student enrols in a Course. Here, enrols in is the relationship between Student and Course.

**Identifier:** An identifier is an attribute that uniquely identifies each record or object of an entity.

Example: For the entity Student, Roll Number can be used as an identifier because each student has a unique roll number.

**Understanding Databases**

Databases store important data required for application operation. They allow applications to manage large data sets effectively. Key database concepts are:

- **Relational Database:** A database that stores data in tables (rows and columns) and links them using relationships.
- **Table:** A collection of related data organised in rows (records) and columns (fields). (as shown in Figure 4.3).
- **Column:** A property or characteristic of an entity (e.g., name, age).
- **Primary Key:** A unique identifier for each record in a table.
- **Foreign Key:** A field that links one table to another by referring to the primary key of another table.

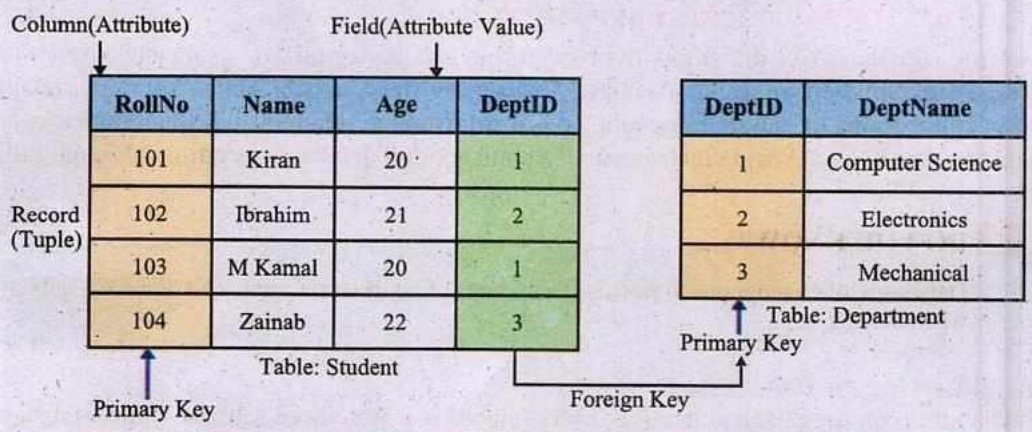



**Figure 4.3: Tables, Columns, Records, Fields, Primary Key, and Foreign Key Relationship**

## Overview of Relational Databases and SQL Structure

Relational databases organize data into tables made of rows and columns. Each table stores related data in a structured manner. Relationships between tables are maintained using keys. SQL stands for Structured Query Language and is used to interact with databases. SQL commands help create, read, update, and delete data.

## Connecting Python to a Database

Connecting Python to a database allows programs to work with stored data. A database connection is required before performing any data operation. Python provides libraries that help establish this connection easily. Once connected, data can be added, retrieved, or modified. Proper connection handling ensures smooth communication between Python and the database.

### Using sqlite3 or MySQL to Establish a Database Connection

The sqlite3 library is built into Python and is used for small databases. It stores data in a single file on the system. MySQL is used for larger and more complex database systems. A connection is created by providing database details such as name and credentials. A successful connection allows Python to access the database.

```
# Import sqlite3 module
import sqlite3
# Connect to the database file named "school.db"
# If the file does not exist, SQLite will create it automatically
connection = sqlite3.connect("school.db")
# Create a cursor object
# Cursor is used to execute SQL queries
cursor = connection.cursor()
# Create "students" table if it does not exist
cursor.execute(""" CREATE TABLE IF NOT EXISTS students
 ( id INTEGER PRIMARY KEY,  name TEXT, age INTEGER, grade TEXT ) """)
 # (Optional) Insert sample data
cursor.execute ("INSERT INTO students (name, age, grade) VALUES ('Ali', 14, '8th')")
cursor.execute ("INSERT INTO students (name, age, grade) VALUES ('Sara', 15, '9th')")
# Save changes
connection.commit()
# This query selects all columns (*) from the table named "students"
cursor.execute("SELECT * FROM students")
# Fetch all records returned by the query
# fetchall() returns a list of rows/records
records = cursor.fetchall()
# Display all records
# Loop through each row in the records list
for row in records:
   # Print each row of the table
   print(row)
# Close the database connection
connection.close()
```

- Connect to Database: Establishes a connection to the school.db SQLite database.
- Create Cursor: Creates a cursor object to execute SQL commands.
- Execute Query: Executes a SQL query to fetch all records from the students table.
- Fetch and Display Data: Fetches all records and prints each row in the result.
- Close Connection: Closes the database connection after the operation.

**Executing SQL Commands From Python**

SQL commands are used to interact with the database from Python. Python sends these commands using a database cursor. Commands can be used to create tables or insert data. Data can also be retrieved using select queries. Executing SQL commands allows Python programs to manage database data effectively.

## CRUD Operations

CRUD stands for Create, Read, Update, and Delete and defines basic database actions. These operations are used to manage data stored in a database (as shown in Figure 4.4 and Table 4.1). Every database-based application depends on CRUD operations. They help maintain accurate and up-to-date information. Python uses SQL commands to perform these operations on databases.

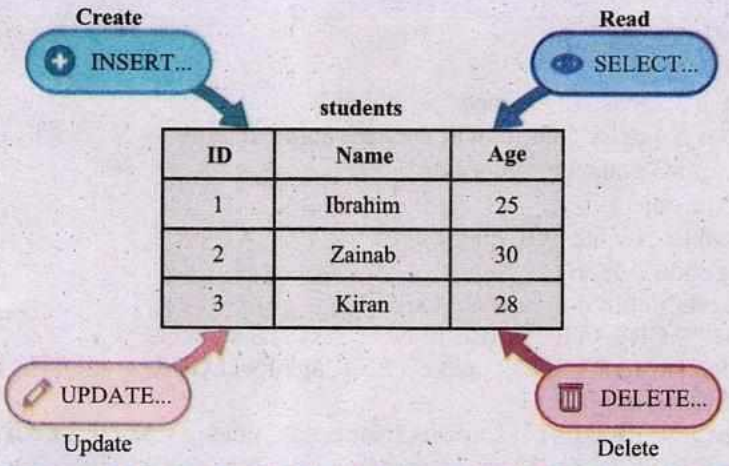



**Figure 4.4: Diagram showing Create, Read, Update, and Delete operations on a table**

**Table 4.1: CRUD operations**

| Command | Command Example | Description |
|---|---|---|
| **INSERT** | INSERT INTO students (ID, Name, Age) VALUES (1, 'John', 20); | Adds a new record to a table. |
| **READ** | SELECT * FROM students; | Retrieves data from a table. |
| **UPDATE** | UPDATE students SET Age = 21 WHERE ID = 1; | Modifies an existing record in a table. |
| **DELETE** | DELETE FROM students WHERE ID = 1; | Removes a record from a table. |

**Create: Adding New Records**

The Create operation is used to add new data to a database table. It stores information such as names, values, or records. SQL insert commands are used for this purpose. Python sends these commands to the database. New records become part of the stored data after execution.

**Read: Retrieving and Displaying Data**

The Read operation is used to fetch data from the database. It helps display stored records when needed. SQL select commands are commonly used for reading data. Python retrieves the results and processes them. This operation does not change the stored data.

**Update: Modifying Existing Records**

The Update operation changes existing data in a database. It is used when stored information needs correction. SQL update commands modify selected records. Python executes these commands based on conditions. Only the specified data is changed during this process.

**CLASS ACTIVITY**

Update a student's marks and display updated data.

## Delete: Removing Data Safely

The Delete operation removes unwanted data from a database. It helps keep the database clean and accurate. SQL delete commands are used with conditions. Python ensures only selected records are removed. Careful use prevents loss of important data.

**Example:**

```
import sqlite3
# Connect to the database
connection = sqlite3.connect("school.db")
# Create a cursor object
cursor = connection.cursor()
# Execute DELETE query (example: delete student with ID = 1)
cursor.execute("DELETE FROM students WHERE ID = 1")
# Save changes
connection.commit()
# Display confirmation
print("Record deleted successfully")
# Close the connection
connection.close()
```

- DELETE FROM students WHERE ID = 1 → removes a specific record
- connection.commit() → saves the changes in database
- Without commit(), deletion will not be permanent

# Summary

**Graphical User Interface**
A Graphical User Interface (GUI) is a visual way of interacting with a computer program using windows, buttons, menus, text boxes, and other graphical elements.

**Tkinter**
Tkinter is Python's built-in library used to create desktop-based graphical user interfaces.

**Window**
A window is the main screen or container in a GUI application where different interface elements are displayed.

**Frame**
A frame is a container used to organize and group widgets inside a GUI window.

**Widget**
A widget is a GUI element, such as a label, button, entry field, menu, or listbox, used to display information or receive input from the user.

**Event-Driven Programming**
Event-driven programming is a programming approach in which the program responds to events or actions performed by the user.

**Relational Database**
A relational database stores data in tables consisting of rows and columns, where related data can be connected using keys.

**SQL**
Structured Query Language (SQL) is a language used to create, manage, and manipulate data in relational databases.

**Create Operation**
The Create operation is used to add new records to a database.

**Read Operation**
The Read operation is used to retrieve and display data from a database.

**Update Operation**
The Update operation is used to modify existing records in a database.

**Delete Operation**
The Delete operation is used to remove records from a database safely.

# EXERCISE

## Multiple Choice Questions

1. The main purpose of a GUI in Python programming is to:
   (a) Create a user-friendly interface for interacting with software
   (b) Manage files in Python
   (c) Handle network operations
   (d) Store and retrieve data from databases

2. Python's built-in GUI toolkit is:
   (a) wxPython (b) Tkinter
   (c) PyQt (d) Kivy

3. The Tkinter widget used to display text is:
   (a) Button (b) Label
   (c) Entry (d) Listbox

4. The pack() method in Tkinter is used to:
   (a) Align widgets vertically or horizontally
   (b) Organize widgets in rows and columns
   (c) Set widget sizes manually
   (d) Create pop-up windows

5. The methods used to organize widgets in Tkinter include:
   (a) grid() (b) pack()
   (c) place() (d) All of the above

6. Event-driven programming in Tkinter means:
   (a) Writing code that runs without user input
   (b) Writing code that responds to user actions such as clicks or key presses
   (c) Writing code to manage database queries
   (d) Writing code to handle network requests

7. The Tkinter widget used to get user input is:
   (a) Label (b) Button
   (c) Entry (d) Listbox

8. The Tkinter layout manager that allows precise positioning of widgets using coordinates is:
   (a) pack() (b) grid()
   (c) place() (d) align()

9. The Tkinter option that connects a button click to a function is:
   (a) bind() (b) configure()
   (c) command= (no parentheses) (d) execute()

10. CRUD operations in databases stand for:
    (a) Create, Read, Update, Delete
    (b) Create, Run, Upload, Delete
    (c) Copy, Retrieve, Upload, Download
    (d) Create, Remove, Update, Manage

## Short Questions

1. What is a GUI and why is it important in application development?
2. What is Tkinter and why is it Python's built-in GUI toolkit?
3. How do you create a window and add frames in Tkinter?
4. Name two common widgets used in Tkinter.
5. What is the purpose of layout management in Tkinter?
6. How does the pack() method organize elements in Tkinter?
7. What is event-driven programming and how is it used in Tkinter?
8. How do you handle user input in Tkinter?
9. What is the CRUD operation in database management?
10. How do you connect Python to a database like SQLite?

## Long Questions

1. Explain the concept of Graphical User Interface (GUI) and discuss its importance in application development.
2. Describe Tkinter as Python's built-in GUI toolkit.
3. What are the key components and widgets in Tkinter?
4. What is layout management in Tkinter, and how are elements organized using pack(), grid(), and place()?
5. How can you design clean and responsive user interfaces in Tkinter?
6. Explain the concept of event-driven programming in Tkinter.
7. How do you handle user input and connect widgets to functions in Tkinter?
8. Discuss the process of connecting Python to a database using SQLite or MySQL.

## Answer Key

1. A
2. B
3. B
4. A
5. D
6. B
7. C
8. C
9. C
10. A

# Chapter 5

# Code Testing and Debugging

**Student Learning Outcomes:**

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

- Use advanced techniques such as unit tests, breakpoints, and watches for testing and debugging Python applications.

## 5.1 Introduction

In this chapter, students will learn the essential techniques of code testing and debugging to ensure their programs function correctly. They will understand the importance of testing for creating reliable applications and explore common types of programming errors and bugs. The chapter covers unit testing using Python's unittest and pytest modules, guiding students on how to write and execute test cases. Students will also be introduced to the process of debugging, including setting breakpoints, using watch expressions, and performing step-by-step debugging. Furthermore, the chapter explains exception handling and how to manage multiple exception types effectively. Finally, students will explore profiling and optimization techniques to measure performance and identify potential bottlenecks in their code.

## 5.2 Testing and Debugging Techniques

Advanced testing and debugging techniques help create reliable and efficient applications. They go beyond basic error fixing to ensure program quality. These techniques identify hidden problems that may affect functionality. Testing and debugging improve performance and user satisfaction. They also make future maintenance easier.

### 5.2.1 Importance of Testing and Debugging

Testing and debugging are essential to ensure applications work correctly under all conditions. They help detect and remove errors before the software is released. Regular testing prevents unexpected failures during operation. Debugging locates the cause of errors and fixes them efficiently. Together, these practices increase reliability and user confidence.

**CLASS ACTIVITY**

Discuss scenarios where errors cause failure, and explain testing importance.

#### Why testing is essential for reliable applications

Testing checks whether an application meets its requirements. It ensures that all functions produce correct results. Testing reduces the risk of crashes or incorrect outputs. It also helps verify that changes do not break existing features. Applications that are thoroughly tested are more trustworthy and stable.

#### Common types of programming errors and bugs

Syntax errors occur when code does not follow language rules. Logic errors happen when the program produces incorrect results. Runtime errors appear while the program is running. Other

common bugs include data errors and resource leaks. Identifying these errors is key to building stable applications (as shown in Figure 5.1).

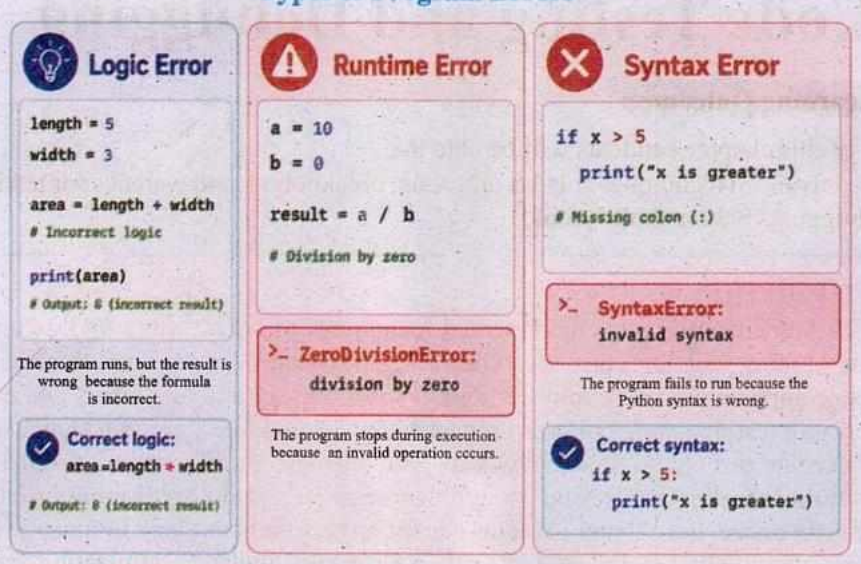



**Figure 5.1: Example showing logic error, runtime error and syntax error**

## Unit Testing

Unit testing is the process of testing individual parts or units of a program to ensure they work correctly. It helps catch errors early and makes debugging easier. Python provides modules like unittest and pytest for writing and running tests. These modules allow automated testing of functions and methods. Unit tests check whether each component produces the expected output. Writing clear and simple test cases improves code reliability.

## Introduction to Python's unittest and pytest modules

The unittest module is built into Python and provides a framework for creating test cases. Pytest is an external module that simplifies writing and running tests. Both modules help verify that code behaves as expected. They can test individual functions, classes, or entire modules. Using these tools improves program stability and reduces errors in larger applications.

**Writing and executing test cases**

Test cases define specific inputs and the expected output for a function. For example, a calculator function can be tested for addition and subtraction results. Python executes these tests and reports failures or errors. Repeated testing ensures that changes in code do not break existing functionality. Automated test execution saves time and improves accuracy in development.

**Example:**

```
# Function to test
# This function takes two input values, a and b.
# It returns the sum of both values.
def add(a, b):
    return a + b
```

```
# This function takes two input values, a and b.
# It returns the product of both values.
def multiply(a, b):
    return a * b

# Using pytest to test functions
# In pytest, every test function usually starts with the word "test_".
# Pytest automatically detects and runs these test functions.

def test_add():
    # This test checks whether the add() function gives the correct result.
    # Here, 2 and 3 are passed as input values.
    # The expected output is 5.
    assert add(2, 3) == 5  # Test passes because 2 + 3 = 5

def test_multiply():
    # This test checks whether the multiply() function gives the correct result.
    # Here, 2 and 3 are passed as input values.
    # The expected output is 6.
    assert multiply(2, 3) == 6  # Test passes because 2 * 3 = 6
```

- pytest simplifies testing. It doesn't require test case classes or boilerplate code like unittest.
- The tests are written as simple functions, and assert is used to check if the output is correct.
- To run the tests, simply execute pytest in the command line in the directory where the file is located.

## Using Breakpoints and Watches

Breakpoints and watches are tools used in debugging to analyze program behavior. They allow pausing program execution and examining data at specific points. These tools help identify errors and understand how code flows. Using breakpoints and watches makes debugging faster and more precise.

## Setting breakpoints in IDEs like PyCharm or VS Code

Breakpoints are markers that pause program execution at a chosen line of code. IDEs like PyCharm and VS Code allow setting breakpoints with a simple click. When the program reaches a breakpoint, execution stops, and the current state can be examined. Breakpoints help focus on specific sections of code without running the entire program.

## Monitoring variable values with watch expressions

Watch expressions track the values of variables while the program is running. They update automatically when variable values change. This helps identify incorrect or unexpected data during execution. Watches allow observing how data changes step by step in real time.

## Step-by-step debugging process

Step-by-step debugging executes code one line or function at a time. It helps follow the program's logic and detect where errors occur. Developers can step into functions, step over lines, or continue execution after a breakpoint. This process makes it easier to locate and fix bugs accurately (as shown in Figure 5.2).

**Example:**

```
def calculate_area(radius):
    # Set a breakpoint here to pause and check the value of 'radius'
    area = 3.14 * radius ** 2
    return area

def main():
    # Step-by-step debugging: starting from here
    radius = 5
    print("Area of circle:", calculate_area(radius))  # Set another breakpoint here to monitor return value

    radius = 10  # Change the radius value
    print("Area of circle:", calculate_area(radius))

# Start the program
if __name__ == "__main__":
    main()
```

- The function calculate_area(radius) calculates the area of a circle using the formula 3.14 * radius ** 2 and returns the calculated value.
- The main() function tests the area calculation twice by using two different radius values: 5 and 10.
- Breakpoints can be placed inside calculate_area() and near the print() statements to check the value of radius, observe the calculated area, and understand step-by-step debugging.

1. **Setting Breakpoints in IDEs like PyCharm:**
    - In PyCharm, you can set breakpoints by clicking the line number next to the code (in the margin).
    - When the program runs and reaches a breakpoint, it pauses. At this point, you can inspect variable values and see the current state of the program.
    - Example: Set breakpoints at radius = 5 and radius = 10 to pause the program execution at these lines.
2. **Monitoring Variable Values with Watch Expressions:**
    - In your IDE, you can use watch expressions to track variables like radius during execution.
    - When you run the debugger, watch expressions will display how the value of radius changes as the program runs.
    - Example: You can add a watch for radius and see its value change from 5 to 10 as the program progresses.
3. **Step-by-Step Debugging Process:**
    - Step-by-step debugging is available in PyCharm. You can choose to:
        - Step Into: Go inside a function to see how it's working (calculate_area in the example).
        - Step Over: Execute the current line and move to the next one without going inside any functions.



Sig5 SideB Process Black 07/17/2025 Computer 12. pecta ctishaq

- Continue: Continue the execution until the next breakpoint.
- Example: You can step over the line area = 3.14 * radius ** 2 to skip calculating the area, or step into calculate_area() to examine how the area is calculated.

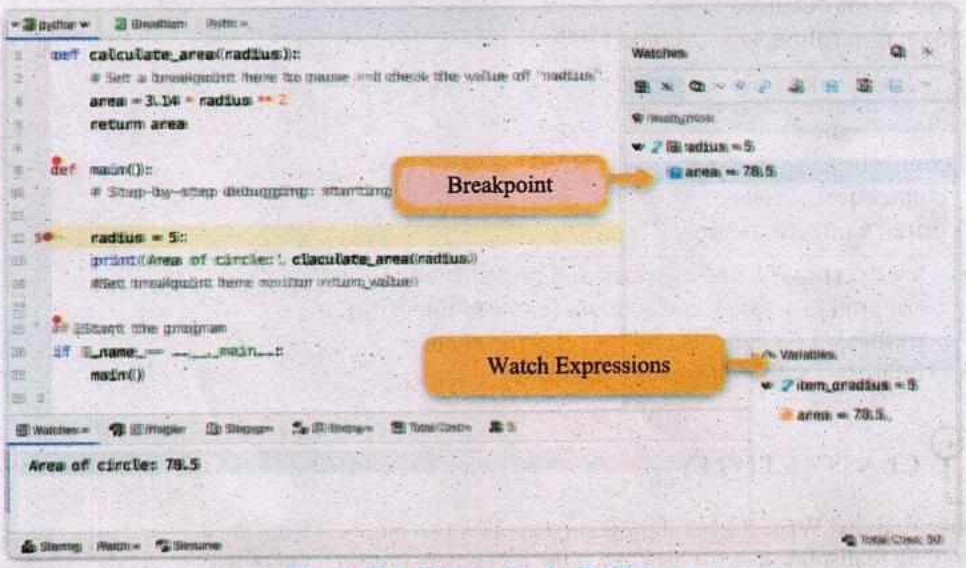



Figure 5.2: Debugging in PyCharm

## Exception Handling

Exception handling is used to manage errors that occur during program execution. It prevents the program from crashing and allows it to respond gracefully to unexpected situations. Proper exception handling improves program reliability and user experience. Python provides built-in tools to catch and handle exceptions efficiently.

**TIDBIT**

Exception handling helps programs manage unexpected problems without stopping suddenly.

**Example:**

```
import sqlite3

try:
    # Connect to database
    connection = sqlite3.connect("school.db")
    cursor = connection.cursor()

    # Insert data into table
    cursor.execute("INSERT INTO students (ID, Name, Age) VALUES (?, ?, ?)", (1, "Ali",
20))

    # Save changes (ensures data integrity)
    connection.commit()
```

```
    print("Data inserted successfully")

except sqlite3.Error as e:
    # Handle database errors
    print("Error occurred:", e)
    connection.rollback()  # Undo changes if error occurs

finally:
    # Close connection
    if connection:
        connection.close()
        print("Connection closed")
```

- try-except → handles errors and prevents crashes
- commit() → saves correct data (ensures integrity)
- rollback() → cancels changes if error occurs
- finally → safely closes the database

**CLASS ACTIVITY**

Class Activity: Write a Python program that takes two numbers from the user and uses exception handling to display an error message if the user enters invalid input or tries to divide by zero.

### Handling multiple exception types effectively

A single program may encounter different types of exceptions. Python allows catching multiple exceptions in one try–except block. Each exception type can be handled with a specific response. This approach prevents unexpected crashes and ensures proper error recovery. Handling multiple exceptions improves program stability and control over errors.

## 5.3 Profiling and Optimization

Profiling and optimization help improve the speed and efficiency of a program. Profiling measures how much time and resources each part of the code uses. Optimization focuses on making slow or resource-heavy parts faster. These techniques are important for building high-performance applications. Regular profiling ensures that code remains efficient as it grows.

**DO YOU KNOW?**

Profiling helps identify slow parts of a program so they can be improved for better performance.

**Example:**

The following code shows a nested loop:

```
for i in range(10000):
    for j in range(10000):
        # Some repeated task
        pass
```

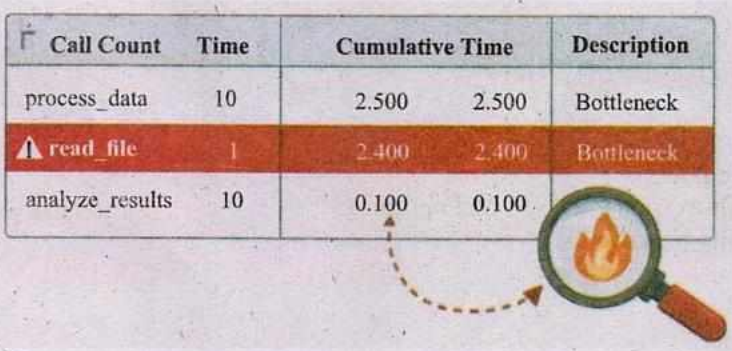
In this code, the outer loop runs 10,000 times. For every single iteration of the outer loop, the inner loop also runs 10,000 times. Therefore, the total number of executions becomes:

$$10000 \times 10000 = 100,000,000$$

This means the inner statement may run one hundred million times, which can make the program very slow. Such a part of the code is called a performance bottleneck. Profiling tools help programmers identify these slow sections so that they can optimise the code and improve program efficiency.

## Measuring performance using profiling tools

Profiling tools track the execution time and memory usage of different functions in a program. Python provides modules like cProfile and timeit for this purpose. These tools generate reports showing which parts of the code are slow or resource-intensive (as shown in Figure 5.3). Measuring performance helps prioritize areas for optimization.



| Call Count | Time | Cumulative Time | | Description |
|---|---|---|---|---|
| process_data | 10 | 2.500 | 2.500 | Bottleneck |
| ⚠ read_file | 1 | 2.400 | 2.400 | Bottleneck |
| analyze_results | 10 | 0.100 | 0.100 | |

Figure 5.3: Diagram showing profiling output highlighting bottlenecks

## Identifying and fixing performance bottlenecks

Performance bottlenecks are sections of code that slow down the program. Profiling reports help locate these bottlenecks. Optimization techniques include improving algorithms, reducing unnecessary operations, or using efficient data structures. Fixing bottlenecks improves overall program speed and responsiveness (as shown in Figure 5.4).

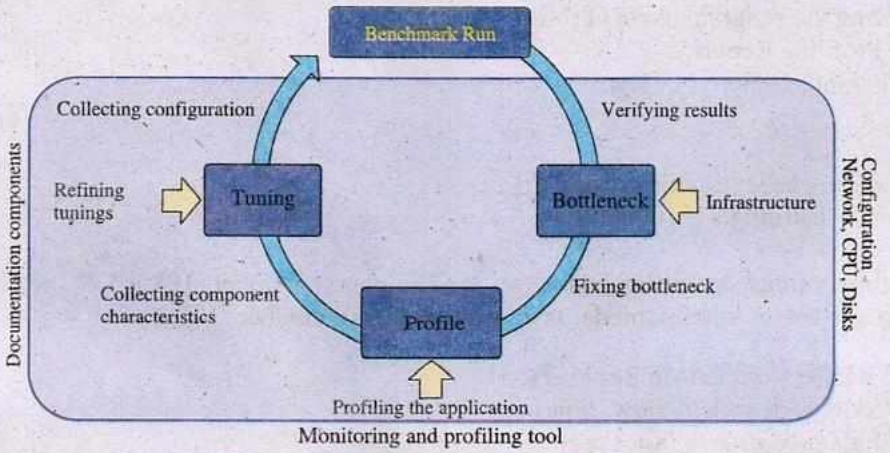



Figure 5.4: Performance Test Cycle

**Example:**

```
import cProfile
import timeit


# Slow code example
def slow_sum(numbers):
    total = 0

    # Adding numbers one by one using a loop
    for num in numbers:
        total = total + num

    return total


# Fast code example
def fast_sum(numbers):
    # Python built-in sum() is faster and optimised
    return sum(numbers)


# Main function
def main():
    numbers = list(range(1, 1000000))

    print("Slow code result:", slow_sum(numbers))
    print("Fast code result:", fast_sum(numbers))


# Profiling the program using cProfile
print("Profiling Report:")
cProfile.run("main()")


# Comparing execution time using timeit
numbers = list(range(1, 1000000))

slow_time = timeit.timeit(lambda: slow_sum(numbers), number=10)
fast_time = timeit.timeit(lambda: fast_sum(numbers), number=10)

print("\nTime Comparison Report:")
print("Slow code time:", slow_time)
print("Fast code time:", fast_time)
```

- cProfile generates a profiling report that shows how much time each function takes during program execution.
- slow_sum() is slower because it adds numbers manually using a loop.
- fast_sum() is faster because it uses Python's built-in sum() function, which is already optimised.
- The function taking more execution time is called the bottleneck.
- timeit helps compare the execution time of slow and fast code.

**Output:** (The exact time values may be different on each computer, but the output will look like this)

```
Profiling Report:
Slow code result: 499999500000
Fast code result: 499999500000

6 function calls in 0.080 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall  function
1       0.060    0.060    0.080    0.080    <module>
1       0.050    0.050    0.050    0.050    slow_sum
1       0.010    0.010    0.010    0.010    {built-in sum}
1       0.010    0.010    0.010    0.010    fast_sum
1       0.000    0.000    0.080    0.080    main

Time Comparison Report:
Slow code time: 0.42
Fast code time: 0.09
```

- Both functions give the same result: 499999500000.
- slow_sum() takes more time because it uses a manual loop.
- fast_sum() takes less time because it uses Python's built-in sum() function.
- Therefore, slow_sum() is the bottleneck in this program.

## Summary

**Testing**

Testing is the process of checking a program to make sure that it works correctly and produces the expected output.

**Debugging**

Debugging is the process of finding and fixing errors or bugs in a program.

**Bug**

A bug is an error, mistake, or fault in a program that causes it to produce an incorrect result or behave unexpectedly.

**Programming Error**

A programming error is a mistake in code that may prevent the program from running correctly. Common errors include syntax errors, logic errors, and runtime errors.

**Unit Testing**

Unit testing is a testing method in which small parts of a programme, such as individual functions, are tested separately to check whether they work correctly.

**Test Case**

A test case is a set of inputs, expected outputs, and conditions used to check whether a program or function is working properly.

**Unittest**

unittest is a built-in Python module used to create and run test cases for checking program correctness.

**Pytest**

pytest is a Python testing framework used to write and execute simple and powerful test cases.

**Breakpoint**

A breakpoint is a point in a program where execution is paused so that the programmer can check the values of variables and understand program behavior.

**Watch Expression**

A watch expression is used during debugging to monitor the value of a variable or expression while the program is running step by step.

**Exception Handling**

Exception handling is a method used to manage runtime errors so that a program does not stop suddenly during execution.

**Profiling**

Profiling is the process of measuring a program's performance to find which parts of the code take more time or use more resources.

**Optimization**

Optimization is the process of improving code so that it runs faster, uses fewer resources, or performs more efficiently.

**Performance Bottleneck**

A performance bottleneck is a part of a program that slows down the overall execution and reduces efficiency.

# EXERCISE

## Multiple Choice Questions

1. Testing is essential for reliable applications because it helps to:
   (a) Optimise code performance
   (b) Identify and fix bugs
   (c) Reduce the program size
   (d) Make code run faster
2. Common types of programming errors include:
   (a) Syntax error
   (b) Logic error
   (c) Runtime error
   (d) All of the above
3. Unit testing focuses on:
   (a) Testing the entire application
   (b) Testing individual components or functions
   (c) Testing the user interface
   (d) Testing database connections
4. The Python module used for unit testing is:
   (a) numpy
   (b) matplotlib
   (c) unittest
   (d) os
5. The main purpose of using testing tools like unittest and pytest:
   (a) to increase program size
   (b) to verify code works correctly
   (c) to design layouts
   (d) to compile programs
6. Breakpoints in debugging are used to:
   (a) Run the program faster
   (b) Pause the program at specific points and inspect its state
   (c) Skip certain code sections
   (d) Optimise the code
7. Watch expressions in debugging are used to:
   (a) Change variable values
   (b) Monitor variable values during program execution
   (c) Pause the program
   (d) Display output messages
8. Step-by-step debugging is used to:
   (a) Run the program faster
   (b) Execute code one line at a time for easier error identification
   (c) Increase the program's performance
   (d) Compile the code
9. The Python keyword pair used for exception handling is:
   (a) try-except
   (b) try-finally
   (c) try-catch
   (d) handle-throw

10. Profiling in programming is used to:
    (a) Improve the user interface design
    (b) Measure performance and identify bottlenecks
    (c) Optimise memory usage only
    (d) Test for security vulnerabilities

## Short Questions

1. Why is testing essential for ensuring reliable applications?
2. What are some common types of programming errors and bugs?
3. What is unit testing, and why is it important in programming?
4. How do Python's unittest and pytest modules help in unit testing?
5. What is the purpose of writing and executing test cases?
6. How do you set breakpoints in IDEs like PyCharm or VS Code?
7. What is the role of monitoring variable values with watch expressions during debugging?
8. What is the step-by-step debugging process in IDEs?
9. How does exception handling help in managing multiple exception types effectively?
10. What tools can be used for profiling and measuring performance in Python?

## Long Questions

1. Why is testing essential for reliable applications, and what role does it play in ensuring program stability?
2. Discuss the common types of programming errors and bugs that occur during software development.
3. What is unit testing, and how do Python's unittest and pytest modules support effective testing?
4. Explain the process of writing and executing test cases using Python's unittest or pytest modules.
5. How do breakpoints and watch expressions help in debugging? Discuss their role in identifying issues in the code.
6. Describe how to set breakpoints in IDEs like PyCharm or VS Code, and explain their significance in the debugging process.
7. What is step-by-step debugging, and how does it help in identifying and fixing bugs in Python programs?
8. How does exception handling work in Python? Explain how to handle multiple exception types effectively to prevent program crashes.

## Answer Key

| | | | |
|---|---|---|---|
| 1. | B | 6. | B |
| 2. | D | 7. | B |
| 3. | B | 8. | B |
| 4. | C | 9. | A |
| 5. | B | 10. | B |

# Chapter 6

# Data and Analysis

**Student Learning Outcomes:**

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

- grasp the interdisciplinary nature of data science, understanding various data types, collection methods, and storage techniques. They will also demonstrate proficiency in data science, machine learning, and data visualization, effectively applying these skills to address real-world business challenges and communicating findings through advanced data visuals.

## Introduction

In this chapter, students will learn the fundamentals of data and analysis and their importance in modern computing. They will explore the basics of data science and understand different types of data, including structured and unstructured data, along with methods of data collection and storage. The chapter introduces machine learning as a way for systems to learn from data and make decisions. The chapter explains how to evaluate model performance using measures like accuracy, precision, recall, and F1-score, and how to improve models through validation and tuning. Finally, the students will be introduced to common tools such as Excel, R, and Python used in data science and machine learning.

## 6.1 Introduction to Data Science

Data science is a field that focuses on collecting, analyzing, and using data to find useful information. It helps convert raw data into meaningful knowledge. Data science uses tools and methods to analyze large amounts of data. It supports decision-making in many areas of life. Data science also helps predict future trends. It is widely used in business, education, and healthcare. Modern technology depends heavily on data science.

**TIDBIT**

Many organizations use data science to support decision making. It helps understand trends and patterns hidden in data.

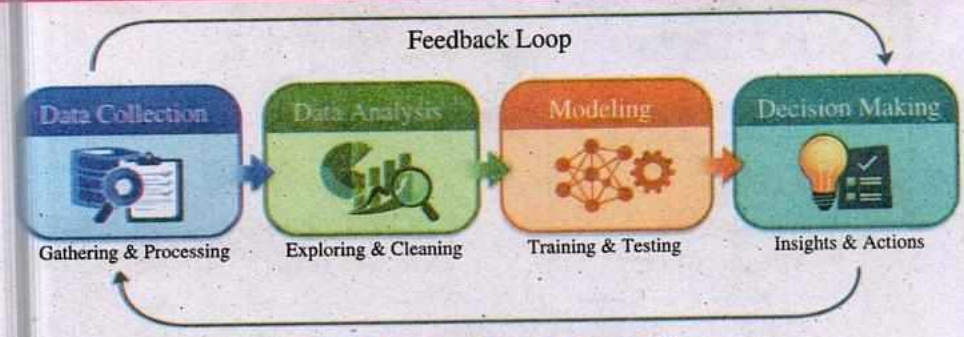



**Figure 6.1: Data Science Work Flow**

Data science means the study of data to discover patterns and useful information. It involves collecting data from different sources. The data is cleaned and organized before analysis. After analysis, the results are interpreted carefully. Data science helps answer questions using facts and data. It reduces guesswork in decision-making. The complete workflow of Data Science is shown in Figure 6.1. Data science turns data into knowledge. This knowledge helps solve real-world problems.

Data science combines knowledge from different subjects. It uses mathematics to work with numbers and calculations. Statistics helps in analyzing and summarizing data. Computer science provides tools and programming skills. Domain knowledge is needed to understand real problems. Communication skills help explain results clearly. Because of this combination, data science is called interdisciplinary. Each field supports the other. Together, they make data analysis more effective.

## 6.2 Understanding Data

Data is the foundation of Data Science. Data is a collection of raw facts and values to produce information. It can be in the form of numbers, text, images, audio or video. Data is produced every day through different activities. Understanding data is important before analysis. Different types of data need different handling methods. Proper data understanding helps in obtaining correct results. Poor data understanding can lead to errors. Data must be accurate and reliable to produce accurate results

**TIDBIT**

Data quality strongly affects final results. Poor data can lead to incorrect conclusions.

### Types of Data: Structured and Unstructured

Structured data is organized in a clear format, such as tables or spreadsheets. It has rows and columns that are easy to understand. Examples include students' mark sheets and databases. Unstructured data does not follow a fixed format (as shown in Figure 6.2). Examples include emails, images, audio, and videos. Unstructured data is difficult to analyze directly. Special tools are needed to process it. Both types of data are important. Modern systems use both structured and unstructured data.

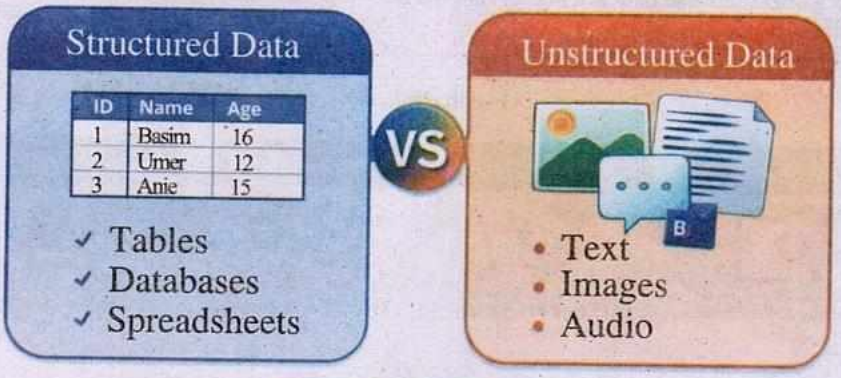



**Figure 6.2: Types of Data**

## Data Sources and Data Collection Methods

Data can be collected from many sources. Common sources include surveys, interviews, forms, sensors, and online platforms (such as blogs, social websites, and others). Manual data collection involves human effort, whereas automatic data collection uses machines and software. Data can be collected in real time or stored over time. The method depends on the purpose of the analysis. Good data sources and accurate data collections improve analysis quality.

## Data Storage and Management Concepts

Data storage refers to saving data for future use. Data can be stored in files, databases, or cloud systems. Proper storage helps protect data from loss. Data management involves organizing and maintaining data. It ensures data is easy to find and use. Security is an important part of data management. Backup systems help prevent data loss. Well-managed data improves efficiency, Data management supports long-term data use.

**DO YOU KNOW?**

Proper understanding of data type is very important before analysis of data.

# 6.3 Overview of Machine Learning

Machine learning is a subfield of Data Science that enables systems to improve from experience without explicit programming. It involves analyzing patterns in data to make predictions or decisions. Large datasets are used to train models, which adapt and improve over time as new data is introduced. This technique is widely applied in modern applications, supporting automation and intelligent systems.

Instead of predefined rules, machine learning allows systems to learn from examples, identifying patterns that guide predictions. As the system processes more data, it becomes more accurate and efficient, powering smart applications and reducing manual work.

## Learning From Data

Learning from data means using past information to improve decisions. In this process, data is given as input to a learning model, which studies relationships in it. Errors are reduced through repeated training, and data becomes more accurate. Learning from data is continuous because the models update when new data is available. This process helps systems become smarter.

## Difference Between Traditional Programming and Machine Learning

Traditional programming uses fixed rules written by programmers. The output is produced by following these rules. Machine learning uses data instead of fixed rules. The system learns rules automatically from data. In traditional programming, changes require new code. In machine learning, changes happen through new data. Machine learning handles complex problems better. It adapts to new situations. Figure 6.3 shows traditional programming vs machine learning

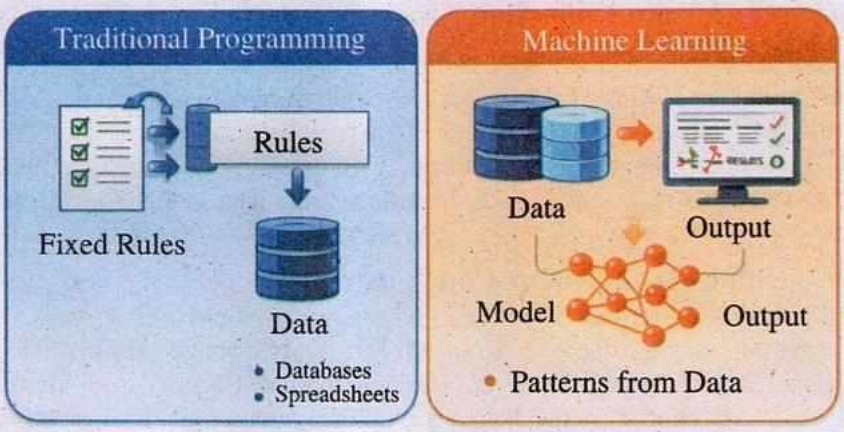



Figure 6.3: Traditional Programming vs Machine Learning

## 6.4 Machine Learning Mechanisms

Machine learning mechanisms describe how systems learn from data. Different mechanisms are used for different problems. Some methods use labeled data, while others do not (as shown in Figure 6.4).

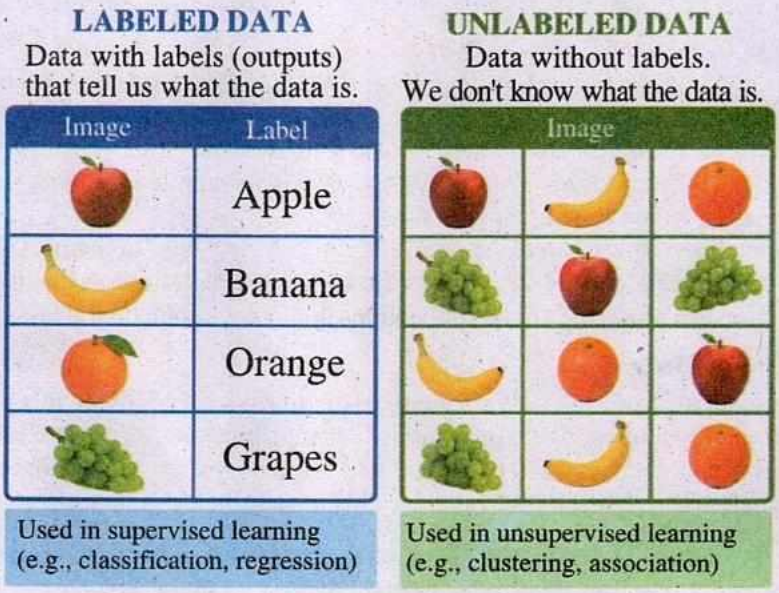



Figure 6.4: Labeled vs Unlabeled Data

Certain methods are learned through feedback and experience. Each mechanism follows a different learning process. Choosing the correct mechanism is important. The quality of learning depends on the method used. These mechanisms support intelligent decision-making. Machine learning systems rely on proper mechanisms to work effectively. Some kinds of machine learning are discussed below:

### Supervised Machine Learning

Supervised machine learning uses labeled data for training. Each data item has an input and a correct output. The system learns by comparing its output with the correct answer. Supervised learning needs large labelled datasets, such as emails labelled as spam or not spam, images labelled as cat or dog, or medical scans labelled as normal or abnormal. Its accuracy usually improves when the model is trained on more high-quality examples. It is widely used in real applications, such as face recognition, disease detection, speech recognition, fraud detection, and recommendation systems. This method is used for prediction tasks. Examples include email spam detection and exam result prediction.

### Unsupervised Machine Learning

Unsupervised machine learning works with unlabeled data. The system finds patterns without guidance. It groups similar data together and makes clusters. This method is useful for discovering hidden structures. No correct output is provided. Learning depends on data similarity. Unsupervised learning helps understand data behavior. It is useful for data exploration. Examples include customer grouping and data clustering.

### Reinforcement Machine Learning

Reinforcement machine learning learns through experience and mimics the human learning. The system interacts with an environment. It receives rewards or penalties for actions. Positive rewards encourage correct behavior. The system improves decision making over time. This method is used in robotics and games. Learning continues through trial and error. The goal is to maximize rewards. Examples of reinforcement learning include a robot learning to navigate a maze through rewards and penalties, AlphaGo improving its strategy by playing Go, an autonomous car optimizing its driving decisions, and a movie recommendation system adjusting its suggestions based on user feedback.

## 6.5 Applications of Machine Learning

Machine learning is used in many areas of modern life. It helps organizations analyze large amounts of data quickly, including Healthcare, Business, Education, and Daily Life (as shown in Figure 6.5). Many systems work automatically using learned patterns. It reduces human effort and saves time.

### Use of Machine Learning in Healthcare

In healthcare, machine learning helps analyze medical data and supports disease detection and diagnosis. For example, it can help detect cancerous cells in medical images, identify early signs of Alzheimer's disease or dementia, and assist in the diagnosis of cardiovascular diseases. Doctors can use these results to plan better and more timely treatments.

**CLASS ACTIVITY**

Provide a simple dataset example, such as marks and grades. Students decide which data can be used for training and testing.

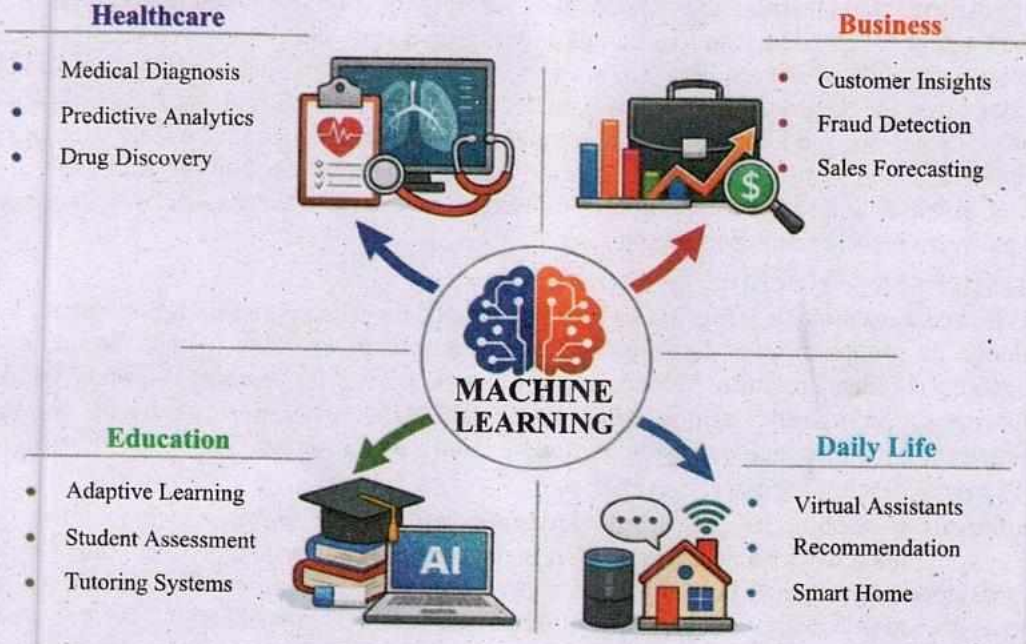



Figure 6.5: Applications of Machine Learning

### Use of Machine Learning in Business

Businesses use machine learning to understand customer behavior. It helps predict sales and market trends. Businesses use it for inventory management. Examples include detecting fraud in financial systems, customer services through robots or chatbots, and suggesting products to customers. Marketing strategies become more effective using machine learning. Business decisions become data-driven. This improves profit and efficiency.

### Use of Machine Learning In Education

In education, machine learning improves learning systems. It helps track students' performance to improve their grades. Personalized learning content is provided. Attendance and assessment systems become smarter. Learning outcomes improve.

### Everyday Applications of Machine Learning

Machine learning is used in daily life activities. Mobile phones use it for face recognition. Voice assistants understand spoken commands. Social media shows personalized content. Email services filter spam messages. Navigation apps suggest routes. Online platforms recommend videos and music. Smart devices use machine learning and daily tasks become easier and faster.

## 6.6 Building a Machine Learning Model

Building a machine learning model involves several clear steps (as shown in Figure 6.6). A simple example of machine learning is student grade prediction. In this example, data about students is first collected, such as attendance, quiz marks, assignment scores, study hours, and

previous exam results. The data is then cleaned and prepared so that it can be used for learning. Important values, also called features, are selected because they help the model understand the factors that affect student performance. After this, the machine learning model is trained using the prepared data. Once trained, the model can predict a student's possible grade based on new input data. This helps teachers identify students who may need extra support before the final examination.

**TIDBIT**

Building a model involves selecting data and features carefully. Training data teaches the model patterns. Testing data checks performance. Feature selection improves results. A well-built model gives reliable predictions.

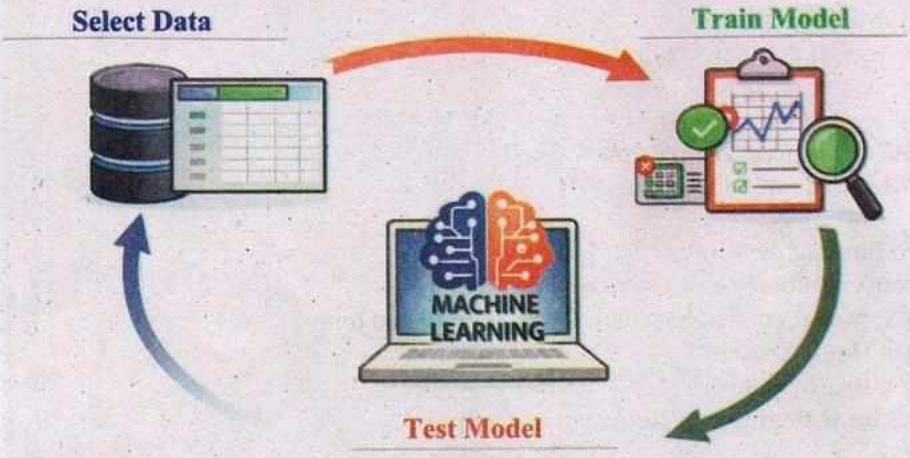



Figure 6.6: Machine Learning Model Building Process

**Example:**

Students' exam scores are often affected by the number of hours they study. The purpose of the following program is to build a simple linear regression model that predicts exam scores based on study hours. The dataset contains examples of hours studied and corresponding exam scores. The data is divided into training and testing sets, the model is trained using the training data, and its performance is evaluated using Mean Squared Error (MSE). Finally, the actual data points and the predicted regression line are plotted to show the relationship between study hours and exam scores.

```
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

```
# Example data (Hours studied vs. Exam scores)
X = np.array([[1], [2], [3], [4], [5], [6], [7], [8], [9], [10]])  # Hours studied

y = np.array([1, 2, 2, 3, 3, 4, 5, 5, 6, 7])  # Exam scores

# Step 1: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 2: Create and train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Step 3: Make predictions
y_pred = model.predict(X_test)

# Step 4: Evaluate the model
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")

# Step 5: Plotting the results
plt.scatter(X, y, color='blue')  # Actual data points
plt.plot(X, model.predict(X), color='red')  # Predicted line
plt.xlabel('Hours Studied')
plt.ylabel('Exam Scores')
plt.title('Linear Regression: Hours vs. Scores')
plt.show()
```

- Data Preparation: The X values represent hours studied, and y values represent the corresponding exam scores (as shown in Figure 6.7).
- Model Training: A linear regression model is created and trained on the training data (X_train, y_train).
- Testing: The model is tested on the test data (X_test) and the mean squared error (MSE) is calculated to evaluate its performance. A lower MSE indicates better model performance because it means the predicted values are closer to the actual values
- Improvement: As more data is gathered or adjusted, the model's accuracy improves.

**Output:**

Mean Squared Error: 0.1484

Model equation: Exam Score = 0.6552 × Hours Studied + 0.1466



SIGO SIDEB Process Black 07/01/2026 12:25 Computer 12. pedagogical

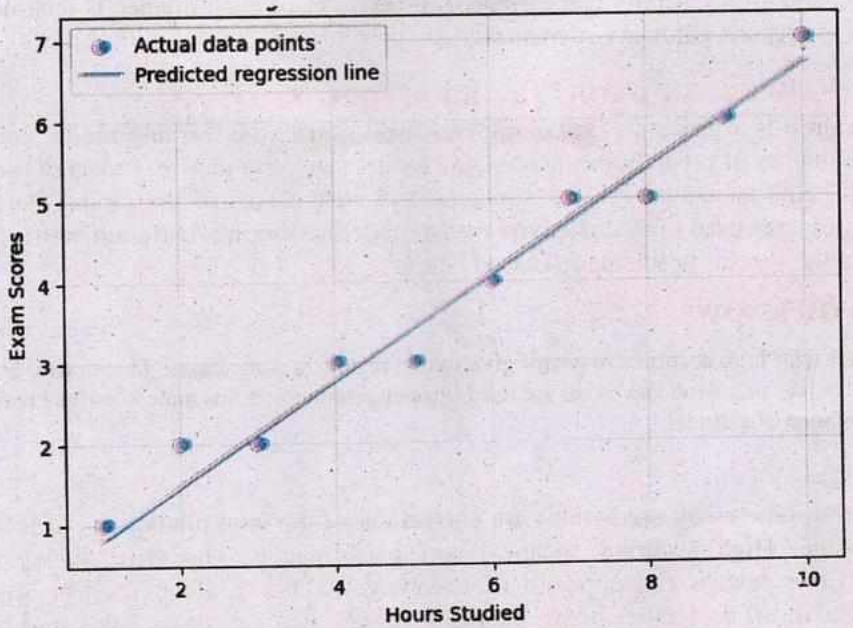



Figure 6.7: Linear Regression: Hours Studied vs Exam Scores

## Selecting Features and Target Values

Features are input values used by the model. Target values are the outputs to be predicted. Correct feature selection improves learning. Unnecessary features can reduce accuracy. Features must be relevant to the problem. Target values must be clearly defined. Example: When predicting student marks, study hours would be the feature, and the target value would be the marks.

## Feature Engineering Basics

Feature engineering means improving data for better learning. It includes selecting, modifying, and creating features. Raw data is often not ready for models. Data may need cleaning and scaling. Feature engineering improves model performance. It helps highlight important information. Good features produce better predictions. This step requires an understanding of data. Feature engineering strengthens the model. Example: In predicting student marks, feature engineering might involve cleaning raw data like study hours, scaling it, and creating new features like "study hours per day" to improve model performance and make better predictions.

## Train-Test Split Of Data

Data is divided into training and testing sets. Training data teaches the model patterns. Testing data checks model performance as it uses unseen data. Results show real performance. This split prevents overfitting. Common splits include 80 percent training and 20 percent testing.

## Building a Simple Predictive Model

A simple predictive model uses basic algorithms to learn from training data. It analyses the data to understand patterns and then makes predictions for new, unseen data. These models are easy

to understand and are especially helpful for beginners. Their performance is measured using test data and the results guide improvements.

## 6.7 Model Evaluation and Performance Metrics

Model evaluation is a process of checking how well a machine learning model performs. It measures the quality of predictions made by the model using test data that was not used during training.This helps in identifying the strengths and weaknesses of the model. Performance metrics are measures used to evaluate how well a model is working. Different metrics are used depending on the type of problem, discussed below:

**DO YOU KNOW?**

A model with high accuracy may still give wrong results in some cases. That is why other measures like precision and recall are used. Proper evaluation helps understand the real performance of a model.

### Accuracy and Error

Accuracy shows how many predictions are correct out of the total predictions. It is measured as a percentage. High accuracy means better performance. The error shows incorrect predictions. Error rate is the opposite of accuracy. Accuracy alone may be misleading sometimes. Balanced data gives better accuracy results. Error analysis helps find mistakes. Both measures are important for evaluation. The accuracy equation is:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- TP (True Positives): Number of correctly predicted positive instances.
- TN (True Negatives): Number of correctly predicted negative instances.
- FP (False Positives): Number of incorrectly predicted positive instances.
- FN (False Negatives): Number of incorrectly predicted negative instances.

### Precision and Recall

Precision measures how many predicted positive results are actually correct. Recall measures how many actual positive cases are correctly identified. Precision focuses on correctness. Recall focuses on completeness. High precision reduces false results. High recall reduces missed cases.

**Precision Equation:**

$$\text{Precision} = \frac{TP}{TP + FP}$$

- TP (True Positives): Correctly predicted positive instances.
- FP (False Positives): Incorrectly predicted positive instances.

**Recall Equation:**

$$\text{Recall} = \frac{TP}{TP + FN}$$

- TP (True Positives): Correctly predicted positive instances.
- FN (False Negatives): Incorrectly predicted negative instances.

### F1-Score

F1-score is the balance between precision and recall. It gives a single performance score. F-measure is useful when the data is unbalanced. It balances correctness and completeness. A higher F-measure means better performance. It avoids relying on accuracy alone. F1-score is commonly used in evaluation. It gives a fair comparison.

**F1-Score Equation:**

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Precision: The proportion of positive predictions that are actually correct.
- Recall: The proportion of actual positive instances that are correctly identified.

**Example:** Suppose we are given the following values:

- TP (True Positives) = 40
- TN (True Negatives) = 50
- FP (False Positives) = 10
- FN (False Negatives) = 5

Now, let's calculate Precision, Recall, and F1-Score based on these values.

- Precision:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{40}{40 + 10} = \frac{40}{50} = 0.8$$

- Recall:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{40}{40 + 5} = \frac{40}{45} = 0.8889$$

- F1-Score:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0.8 \times 0.8889}{0.8 + 0.8889} = 2 \times \frac{0.7111}{1.6889} = 0.8421$$

- Final Results:
- Precision = 0.8
- Recall = 0.8889
- F1-Score = 0.8421

## 6.8 Validation and Model Improvement

Validation is used to check how well a model works on new data. It is done after training and testing to confirm model reliability. It prevents models from memorizing data. Validation supports fair evaluation. Model improvement uses feedback from validation results. Weak areas are identified and corrected. Validation is important before real use.

**TIDBIT**

Validation checks how a model works on new data. It prevents overfitting. Model improvement uses tuning and adjustments. Good validation increases reliability. Strong models perform well in real conditions.

### Validation Data Set Preparation

A validation dataset is separate from training and testing data that is used to check model performance. Validation data must be clean and accurate to represent real conditions. The model does not learn from validation data. Validation helps detect overfitting and improves evaluation. Balanced data gives better results. This step increases model confidence.

### Model Robustness and Reliability

Robustness means a model performs well in different situations. A reliable model gives consistent results. Robust models handle noise in data. They do not fail easily with small changes. Reliability is important for real-world use. Testing under different conditions checks robustness. Strong models maintain accuracy. Reliability builds trust in systems. Robust models are more useful.

### Improving Models Using Hyperparameter Tuning

Hyperparameter tuning adjusts model settings. These settings control learning behavior. It helps improve accuracy. It reduces errors in predictions. Parameters are tested with different values. Validation data guides tuning. Proper tuning improves performance. Poor tuning reduces effectiveness.

### Simple Analogy: Adjusting the Volume of a Speaker

Hyperparameter tuning is like adjusting the volume of a speaker. If the volume is too high, the sound becomes distorted and unclear. If the volume is too low, the sound cannot be heard properly. However, when the volume is set at the right level, the sound becomes clear and balanced.

Similarly, in machine learning, if hyperparameters are not set properly, the model may perform poorly. The correct hyperparameter values help the model learn better and produce more accurate results.

## 6.9 Predictive Modeling and Causality

Predictive modeling uses data to predict future outcomes. Machine learning models analyze past data patterns. These patterns help estimate what may happen next. Predictive models are used in many fields. They support planning and decision making. Predictions are based on probability. They do not always explain reasons. Understanding causality is also important. Both concepts are used together in analysis.

**CLASS ACTIVITY**

Give examples like weather prediction. Ask students whether prediction explains the cause. Discuss the difference.

### Predictive Outcomes in Machine Learning

Predictive outcomes are results produced by a trained model. The model uses past data to make predictions. These outcomes estimate future values or events. Prediction accuracy depends on data quality. Good training improves predictions. For example, businesses use predictions for forecasting the market or product selling trends. Healthcare uses predictions for diagnosis. Predictions support better decisions.

### Understanding Causality

Causality explains why an event happens. It shows a cause-and-effect relationship. One factor directly influences another. Causality is important in scientific studies to understand real reasons behind events. Data alone may not show causality but experiments often identify causes. Causal understanding improves decisions and avoids false assumptions.

### Difference Between Prediction and Causation

Prediction estimates what may happen based on historical data. It focuses on forecasting outcomes, where the model uses past information to estimate future values or events. For example, a model may predict a student's final grade by analysing previous marks, attendance, and study hours. Similarly, a weather model may predict tomorrow's temperature by using past weather records. A model can make accurate predictions without necessarily understanding the underlying causes. This makes prediction useful in areas such as sales forecasting, stock trend analysis, disease risk prediction, and student performance estimation.

Causation, on the other hand, explains why something happens by identifying cause-and-effect relationships between variables. It seeks to uncover the reasons behind events and how one factor directly influences another. For example, if regular study hours directly improve exam performance, then study time may be considered a cause of better grades. Similarly, if smoking directly increases the risk of lung disease, this shows a causal relationship. Understanding causality is important for decision-making because it helps avoid false assumptions and provides deeper insight into real-world problems.

## 6.10 Machine Learning Tools and Platforms

Machine learning uses different tools for data analysis and modeling. Tools help collect, process, and analyze data. Some tools are simple and beginner-friendly, while others support advanced machine learning tasks. Choosing the right tool is important. Each tool has specific strengths. Tools improve efficiency and accuracy and build strong foundations. These platforms support modern data science work. Machine learning tools and platforms include:

- Excel for data analysis
- R for data science
- Python and Jupyter notebooks for machine learning

### Using Excel for Data Analysis

Excel is widely used for basic data analysis by organizing the data in rows and columns. Excel supports sorting, filtering, and data visualization using charts. Charts help visualize data. Simple formulas perform calculations. Excel is easy to learn and is useful for small datasets. Beginners often start with Excel, and it supports early data understanding.

## Introduction to R for Data Science

R is a programming language for data analysis that is widely used in research. It provides strong graphical tools to visualize data. R handles large datasets efficiently. Many packages support data science tasks. Learning R improves analytical skills that support machine learning basics.

## Python and Jupyter Notebooks for Machine Learning

Python is a popular programming language for machine learning. It has many useful libraries such as Pandas, NumPy, and others Python supports data analysis and modeling. Jupyter notebooks provide an interactive environment. Code and results appear together on Jupyter notebook. This helps learning and testing the code on an interactive Integrated Development Environment (IDE). Python supports advanced machine learning and is widely used in industry.

# Summary

**Data Science**
Data science is the process of collecting, organising, analysing, and interpreting data to find useful information and support decision-making.

**Structured and Unstructured Data**
Structured data is arranged in a fixed format, such as tables with rows and columns. Unstructured data does not follow a fixed format, such as images, videos, emails, and audio files.

**Machine Learning**
Machine learning is a branch of artificial intelligence in which computers learn patterns from data and use them to make predictions or decisions.

**Supervised Learning**
Supervised learning is a machine learning method in which the model learns from labelled data, such as study hours and known exam scores.

**Unsupervised Learning**
Unsupervised learning is a machine learning method in which the model finds hidden patterns or groups in data without labelled outputs.

**Reinforcement Learning**
Reinforcement learning is a learning method in which an agent learns by taking actions and receiving rewards or penalties.

**Feature and Target Value**
A feature is an input value used by a model, such as study hours or attendance. A target value is the output the model tries to predict, such as an exam score.

**Train-Test Split**
Train-test split is the process of dividing data into training data for teaching the model and testing data for checking its performance.

**Predictive Model**
A predictive model uses past data to estimate future values or outcomes.

**Model Evaluation**
Model evaluation is the process of checking how well a machine learning model performs using measures such as accuracy, error, precision, recall, and F1-score.

**Parameter Tuning**
Parameter tuning is the process of adjusting model settings to improve its performance.

**Prediction**
Prediction is the process of estimating a future value or event based on past data.

**Causality**
Causality explains cause-and-effect relationships between variables and helps identify why something happens.

**Machine Learning Tools**
Machine learning tools are software platforms used to analyse data and build models. Common examples include Excel, R, Python, and Jupyter Notebook.

# EXERCISE

## Multiple Choice Questions

1. Data science is mainly concerned with:
   (a) Designing hardware (b) Extracting useful information from data
   (c) Writing only code (d) Creating websites
2. Data organised in rows and columns is called:
   (a) Unstructured data (b) Structured data
   (c) Raw data (d) Random data
3. An example of unstructured data is:
   (a) Tables (b) Spreadsheets
   (c) Images and videos (d) Databases
4. Machine learning allows systems to:
   (a) Work without data (b) Learn from data and improve performance
   (c) Only store data (d) Replace databases
5. The type of machine learning that uses labelled data is:
   (a) Unsupervised learning (b) Reinforcement learning
   (c) Supervised learning (d) Rule-based learning
6. The machine learning method that learns through rewards and penalties is:
   (a) Supervised learning (b) Unsupervised learning
   (c) Reinforcement learning (d) Rule-based learning
7. Feature engineering is used to:
   (a) Delete data (b) Improve data for better model performance
   (c) Store data (d) Visualise data
8. The metric that measures the overall correctness of a model is:
   (a) Precision (b) Recall
   (c) Accuracy (d) F1_score
9. Train -test split is used to:
   (a) Delete data (b) Divide data for training and testing
   (c) Store data (d) Visualise data
10. A commonly used tool for machine learning and data analysis is:
    (a) MS Word (b) Python
    (c) PowerPoint (d) Notepad

## Short Questions

1. What is data science?
2. What are the two main types of data?
3. What is the difference between structured and unstructured data?
4. Name two common data collection methods.
5. What is machine learning?

6. What is supervised learning?
7. What is the purpose of feature selection in machine learning?
8. What does accuracy measure in a model?
9. What is the difference between prediction and causality?
10. Name any one tool used for machine learning.

## Long Questions

1. Explain the concept of data science and discuss its importance in modern applications.
2. Describe different types of data and explain various data sources and collection methods.
3. Explain the concept of machine learning and describe its different mechanisms, including supervised, unsupervised, reinforcement, and rule-based learning.
4. Discuss the applications of machine learning in healthcare, business, education, and daily life.
5. Explain the steps involved in building a machine learning model, including feature selection, feature engineering, and train-test split.
6. Describe model evaluation techniques and explain accuracy, precision, recall, and F1-score.
7. Explain validation in machine learning and discuss how models can be improved using parameter tuning.
8. Discuss predictive modeling and causality, and explain the importance of model interpretation in real-world use cases.

## Answer Key

1. B
2. B
3. C
4. B
5. C
6. C
7. B
8. C
9. B
10. B

# Chapter 7

# Hypothesis Testing

**Student Learning Outcomes:**

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

- Learn how to form hypotheses and perform hypothesis testing.
- Learn how to communicate findings using advanced data visuals and tie them back to hypotheses.

## Introduction

In this chapter, students will learn the concept of hypothesis testing and its role in data analysis. They will understand what a hypothesis is and how research questions are converted into testable hypotheses. The chapter introduces different types of hypotheses, including the null hypothesis ($H_0$), and explains how to formulate correct hypotheses. Students will explore key concepts such as test statistics, critical regions, and p-values, and learn their importance in decision-making. They will follow the step-by-step process of hypothesis testing through simple numerical examples. The chapter also highlights the role of data visualization in presenting and linking results with hypotheses. In addition, students will learn about ethical issues such as bias and responsible use of data. By the end of this chapter, students will be able to interpret results and communicate conclusions clearly.

## 7.1 Introduction to Hypothesis Testing

Hypothesis testing is a method used to make decisions using data. It checks whether an assumption is true or false. This method is widely used in research and analysis. Hypothesis testing follows a clear process that helps accept or reject assumptions. Data is collected and analyzed carefully. This method reduces guesswork as it supports evidence-based decisions.

**TIDBIT**

Hypothesis testing checks assumptions using data. It helps decide whether a claim is true. This method is used in research and experiments. It reduces guesswork. Results depend on proper testing.

### Meaning of Hypothesis

A hypothesis is an assumption or statement in research that is made before testing begins. The statement can be tested using data. A hypothesis predicts an outcome; therefore, must be clear and measurable. Data is used to check it. A correct hypothesis guides analysis. It helps focus the study.

$h\theta(x)=y$

**Where:**

- $h\theta(x)$: The hypothesis function with parameters ( $\theta$) applied to input ( x ).
- ( x ): The input features or data points (e.g., attributes of a dataset).

- ( y ): The predicted output or target value (the model's prediction for the given input).
- ( θ): Parameters or weights of the model (for example, coefficients in a linear regression model).

### Research Questions and Hypotheses

Research questions define what is being studied. A hypothesis is formed from the research question. Questions guide data collection. Hypotheses provide possible answers. Clear questions lead to clear hypotheses. Both are closely connected. Good research starts with good questions. Hypotheses make the analysis focused. They improve study direction.

### Role of Hypothesis Testing in Data Analysis

Hypothesis testing helps analyze data logically to check whether results are meaningful. Testing avoids random conclusions. It compares results with expectations. Hypothesis testing supports accuracy, which is used in experiments and surveys. Decisions are based on evidence. This method improves confidence in results.

## 7.2 Types of Hypotheses

Hypotheses are classified into different types. Each type has a specific purpose. The most common types are null and alternative hypotheses. These types work together in testing. Correct hypotheses improve analysis quality by reducing confusion. They guide decision-making. Understanding hypothesis types is important. They form the base of hypothesis testing.

**CLASS ACTIVITY**

Students write a null and alternative hypothesis for marks or percentage situation. Discuss correctness.

### Null Hypothesis ($H_0$)

The null hypothesis states that there is no effect or difference. It is written as a default assumption. Data is tested against this hypothesis. If evidence is strong, the null hypothesis is rejected. It is usually represented by $H_0$. The null hypothesis helps avoid bias. It provides a comparison base. Many tests start with this hypothesis. It is important in decision-making.

### Alternative Hypothesis ($H_1$ or $H_a$)

The alternative hypothesis states that there is an effect or difference. It is written as opposite of the null hypothesis. It is represented by $H_1$ or Ha. The goal of testing is often to support this hypothesis. Evidence must be strong to accept it. It represents change or impact. Alternative hypotheses guide research direction. They explain expected outcomes. Testing compares both hypotheses.

**Example:**

**Sample Research Questions and Hypotheses:**

**Research Question:**

Does the amount of study time affect students' exam scores?

**Hypothesis:**

$H_0$ (Null Hypothesis): There is no significant relationship between study time and exam scores.

$H_1$ (Alternative Hypothesis): The amount of study time significantly affects students' exam scores.

### Formulating correct hypotheses

Correct hypotheses must be clear and specific. They should be testable using data. Vague statements reduce accuracy. Hypotheses must match research questions. Variables should be defined clearly. Simple language improves understanding. Correct formulation saves time. It leads to reliable results. Proper hypotheses improve analysis quality.

## 7.3 Concepts in Hypothesis Testing

Hypothesis testing uses several important concepts to make correct decisions. Each concept has a specific role in testing. Understanding them avoids confusion. They guide acceptance or rejection of hypotheses. These concepts are used together. Proper knowledge improves accuracy. They are applied in data analysis. Learning these concepts is essential.

### Test Statistics

Test statistics are numerical values. They are calculated from sample data. These values summarize data behavior. Test statistics help compare results and support decisions. They show how far results are from expectations. Different tests use different statistics. Examples include t-value and chi-square value.

**T-value Test:**

$$t = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}}$$

**Where:**

- $\bar{x}$: Sample mean
- $\mu_0$: Population mean under the null hypothesis
- s: Sample standard deviation
- n: Sample size

**Example:**

Suppose we have a sample of 25 students with an average score of 80, and we want to test if the mean score is significantly different from the population mean of 75. The sample standard deviation is 10.

$$t = \frac{80-75}{\frac{10}{\sqrt{25}}} = \frac{5}{\frac{10}{5}} = \frac{5}{2} = 2.5$$

**Chi-Square Test:**

$$x^2 = \sum \frac{(o_i - E_i)^2}{E_i}$$

**Where:**

- $O_i$ : Observed frequency in each category
- $E_i$ : Expected frequency in each category

**Example:**

Suppose you roll a fair die 60 times and observe the following frequencies:

| Face of Die | Observed Frequency ($(O_i)$) | Expected Frequency ($(E_i)$) |
|---|---|---|
| 1 | 10 | 10 |
| 2 | 12 | 10 |
| 3 | 8 | 10 |
| 4 | 14 | 10 |
| 5 | 9 | 10 |
| 6 | 7 | 10 |

$$x^2 = \frac{(10-10)^2}{10} + \frac{(12-10)^2}{10} + \frac{(8-10)^2}{10} + \frac{(14-10)^2}{10} + \frac{(9-10)^2}{10} + \frac{(7-10)^2}{10}$$

$$x^2 = 0 + \frac{4}{10} + \frac{4}{10} + \frac{16}{10} + \frac{1}{10} + \frac{9}{10}$$

$$x^2 = 0 + 0.4 + 0.4 + 1.6 + 0.1 + 0.9 = 3.4$$

Now, the calculated $x^2$ value is 3.4, and we would compare this value against the critical value from the Chi-Square distribution table at a certain significance level (e.g., $\alpha = 0.05$) and degrees of freedom to make a decision.

## Critical Region

The critical region is a range of values. If test statistics fall in this region, the null hypothesis is rejected. It is defined before testing. The region depends on the significance level. It shows extreme outcomes. Critical regions help make decisions. They reduce uncertainty. Graphs often show critical regions. Understanding this concept is important.

### Significance Level

The significance level shows risk of error. It is written as alpha. Common values include 0.05. It defines how strict a test is. Lower values mean stricter testing. It is chosen before analysis. Significance level controls false results and affects decision-making. Proper choice improves reliability.

### P-value and its Importance

The p-value shows the strength of evidence against the null hypothesis. It measures the probability of getting the observed results, or more extreme results, if the null hypothesis is true. A small p-value supports the rejection of the null hypothesis because it suggests that the result is unlikely to have occurred by chance. The p-value is compared with the significance level, such as (0.05), to make a decision.

For example, suppose a teacher wants to test whether a new teaching method improves students' marks. The null hypothesis states that the new method has no effect. After analysing the test scores, the p-value is found to be (0.03). Since ($0.03 < 0.05$), the null hypothesis may be rejected. This means there is statistical evidence that the new teaching method may have improved students' performance.

Similarly, in medicine, researchers may test whether a new drug is more effective than an old drug. If the p-value is (0.01), it gives strong evidence that the improvement is unlikely to be due to chance alone.

P-values help simplify statistical decisions and support data-driven conclusions. A clear understanding of p-values helps avoid mistakes, such as rejecting or accepting a hypothesis without proper evidence.

## 7.4 Performing Hypothesis Tests

Performing hypothesis tests follows a clear process (as shown in Figure 7.1). The process starts with defining hypotheses. Data is collected carefully. Test statistics are calculated. Results are compared with criteria. A decision is then made. Proper steps reduce errors. This process ensures fair testing. Hypothesis testing supports reliable conclusions.

**TIDBIT**

Testing follows clear steps. Data is collected and analyzed. Results are compared with criteria. Conclusions are drawn carefully. Correct tests give reliable outcomes.

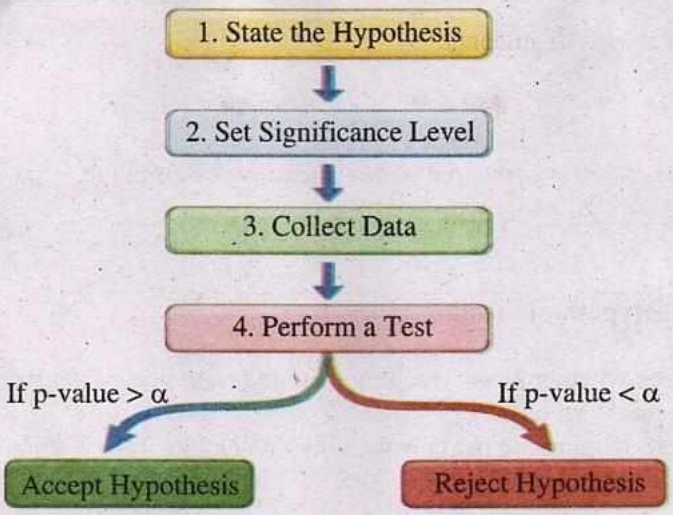



Figure 7.1: Hypothesis Testing Process

## Steps in Hypothesis Testing

First, null and alternative hypotheses are defined. Next, a significance level is selected. Data is collected from samples. Test statistics are calculated. The p-value is found. Results are compared with the significance level. A decision is made to accept or reject.

## Simple Hypothesis Testing Examples

Simple examples help understand testing. One example is testing exam results. Another example is checking average height. Data is collected from samples. Results are analyzed using basic tests. Conclusions are drawn clearly. Simple examples build confidence. They explain concept easily.

## Advanced Tests: F-Test and Chi-Square Test

Advanced tests handle complex data. The F-test compares variances. It is used in analysis of variance. The Chi-Square test checks relationships. It is used with categorical data. These tests support deeper analysis. They follow strict rules. Understanding them improves skills. Advanced tests expand knowledge.

**F-test Example:**

$$F = \frac{S_1^2}{S_2^2}$$

**Where:**

- $S_1^2$ = Variance of sample1
- $S_2^2$ = Variance of sample 2

**Example:**
Suppose we have two groups of students:
Sample 1 variance = 16
Sample 2 variance = 9

F=16/9=1.78

The calculated F-value (1.78) is compared with a critical value from the F-table to decide whether the variances are significantly different.

**Example:**

**Numerical Example: Hypothesis Testing**

A school wants to check whether a new teaching method has improved students' marks.
The previous average marks were 70.
A sample of 10 students taught with the new method obtained the following marks:
72, 75, 78, 74, 77, 80, 76, 79, 81, 78

**Step 1: State the Hypotheses**

**Null Hypothesis ($H_0$):** The new method does not improve marks ( $\mu = 70$)
**Alternative Hypothesis ($H_1$):** The new method improves marks ( $\mu > 70$))

**Step 2: Set Significance Level**
$\alpha = 0.05$

**Step 3: Calculate Sample Mean**

$$\bar{x} = \frac{Sum\,of\,all\,values}{n} = \frac{770}{10} = 77$$

**Step 4: Calculate Standard Deviation**

$$s = \sqrt{\frac{\sum\left(x_i - \bar{x}\right)^2}{n-1}} = \sqrt{\frac{70}{9}} \approx 2.79$$

**Step 5: Calculate t-value**

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}} = \frac{77 - 70}{2.79/\sqrt{10}} \approx \frac{7}{0.88} \approx 7.93$$

**Step 6: Decision Rule**
**df=n-1=10-1=9**
Using the t-distribution table or statistical software, the p-value for $t = 7.93$and $df = 9$ is approximately: p = 0.00002
For (df = 9), the **p-value is very small** ($p < 0.05$)

**Step 7: Conclusion**
Since
p-value < α
We reject the null hypothesis.
**Final Conclusion**
The new teaching method significantly improves students' marks.

## 7.5 Data Visualization For Hypothesis Testing

Data visualization plays an important role in hypothesis testing by presenting data in a clear and understandable form. It helps researchers observe patterns, trends, and relationships in data before applying statistical tests. Visual representations make complex data easier to interpret and reduce the chances of misunderstanding results. Through charts and graphs, differences between groups can be easily compared (as shown in Figure 7.2). Visualization also assists in identifying outliers and anomalies in datasets. It supports better decision-making by providing visual evidence. Overall, data visualization strengthens the process of hypothesis testing by improving clarity and accuracy.

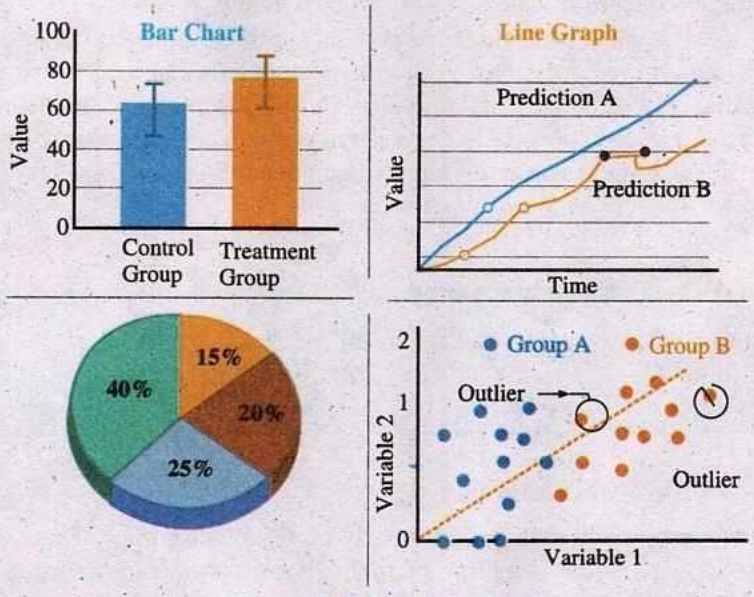



Figure 7.2: Data Visualization

### Importance of Data Visualization

Data visualization is important because it converts raw data into meaningful visual forms. It helps researchers quickly understand large datasets that are difficult to analyze numerically. Visual tools highlight trends, distributions, and variations in data clearly. They make communication of results easier to both technical and non-technical audiences. Visualization also reduces errors by making unusual patterns visible. It supports better interpretation of

statistical findings. Therefore, data visualization is an essential step in data analysis and hypothesis testing.

### Graphs and Charts for Presenting Results

Graphs and charts are commonly used tools for presenting results in hypothesis testing. **Bar charts** are useful for comparing categories or groups. Line graphs show trends and changes over time. **Pie charts** represent proportions and percentages effectively. **Scatter plots** display relationships between two variables. These visual tools make results more attractive, clear, and easier to interpret (as shown in Figure 7.2).

### Linking Data Visuals With Hypotheses

Data visuals help in directly linking results with the proposed hypothesis. They allow researchers to visually check whether the data support or reject the hypothesis. For example, a comparison graph can show differences between control and experimental treatment groups. Trends in line graphs can confirm predicted outcomes. Scatter plots help identify relationships stated in the hypothesis. Visual evidence strengthens logical reasoning in hypothesis testing. Thus, data visualization provides strong support in validating or disproving hypotheses.

## 7.6 Ethical Issues in Data Science and Analysis

Ethical issues are a major concern in data science and data analysis. As data is collected and analyzed on a large scale, it must be handled responsibly. Ethical practices ensure fairness, accuracy, and trust in data-driven decisions. Poor ethical standards can lead to the misuse of data and harm individuals or groups. Data scientists must follow ethical guidelines while collecting, processing, and interpreting data. Transparency and accountability are essential in every stage of data analysis. Ethical awareness helps maintain public confidence in data science applications.

**? DO YOU KNOW?**

Biased data can produce unfair results. Ethical data science protects privacy and avoids discrimination. Responsible data use builds trust in technology.

### Bias in Data Collection and Analysis

Bias in data collection occurs when data does not fairly represent all groups. This can happen due to limited data sources or unfair sampling methods. Biased data leads to inaccurate and misleading analysis results. During analysis, personal assumptions may also introduce bias. Such bias can affect decision-making and reinforce unfair outcomes. Identifying and mitigating bias is essential for reliable results. Therefore, careful data collection and objective analysis are necessary to ensure fairness. Here are simple examples of bias in data collection and analysis:

1. Survey Bias: A survey about student performance is conducted only in top-performing schools, ignoring average or low-performing schools.

2. Sampling Bias: A health study includes only young adults, so results do not represent older people.
3. Gender Bias: A hiring dataset contains mostly male candidates, leading the model to favour male applicants.
4. Geographical Bias: Data collected only from urban areas is used to make decisions for rural populations.
5. Confirmation Bias: A researcher focuses only on data that supports their belief and ignores contradictory results.

### Ethical Use of Data and Models

Ethical use of data involves respecting privacy and confidentiality of individuals. Data should be collected with proper consent and used only for intended purposes. Machine learning models must be designed to avoid discrimination. Sensitive data should be protected from unauthorized access. Ethical guidelines help prevent misuse of predictive models. Transparency in model design builds trust among users. Responsible use of data and models ensures positive social impact.

### Responsible Communication of Findings

Responsible communication of findings is essential in data science. Results should be presented honestly without exaggeration or manipulation. Limitations of the data and analysis must be clearly stated. Misleading visuals or selective reporting should be avoided. Clear explanations help stakeholders understand the results accurately. Ethical communication prevents misinterpretation and misuse of findings. Thus, truthful reporting strengthens credibility and trust in data analysis.

## 7.7 Communicating Results and Conclusions

Communicating results and conclusions is an important step in data analysis and research. It helps readers understand what the data shows and what decisions can be made. Clear communication ensures that results are not misunderstood or misused. Researchers must explain findings in a simple and logical manner. Conclusions should be based strictly on analyzed data. Proper communication improves transparency and credibility. Effective reporting also supports future research and decision-making.

### Interpreting Test Results

Interpreting test results involves understanding what statistical outcomes mean. It includes analyzing values such as p-values, confidence levels, or test statistics. Results help determine whether a hypothesis is accepted or rejected. Researchers must consider the context of the data while interpreting results. Misinterpretation can lead to incorrect conclusions. Clear explanation of results is essential for accuracy. Proper interpretation ensures reliable and meaningful findings.

## Presenting Findings Clearly

Presenting findings clearly helps readers easily understand research outcomes. Results should be organized in a logical and structured way. Tables, charts, and graphs can be used to support explanations. Simple language should be used instead of technical jargon. Key points must be highlighted for better understanding. Clear presentation reduces confusion and improves communication. Well-presented findings make the research more effective and useful.

## Connecting Conclusions to Hypotheses

Conclusions must be directly linked to the original hypotheses. Researchers should clearly state whether the hypothesis is supported or not. Data and test results should justify the conclusion. Personal opinions should not influence the final conclusion. Clear connections strengthen the validity of research. This step ensures logical consistency in the study. Connecting conclusions to hypotheses completes the research process effectively.

## Summary

**Hypothesis**
A hypothesis is a testable statement or assumption about a population, situation, or relationship between variables.

**Hypothesis Testing**
Hypothesis testing is a statistical method used to decide whether there is enough evidence in sample data to support or reject a claim.

**Null Hypothesis**
The null hypothesis, written as (H_0), states that there is no significant difference, effect, or relationship between variables.

**Alternative Hypothesis**
The alternative hypothesis, written as (H_1) or (H_a), states that there is a significant difference, effect, or relationship between variables.

**Test Statistic**
A test statistic is a numerical value calculated from sample data and used to make a decision in hypothesis testing. Examples include (z), (t), and (F) values.

**Critical Region**
The critical region is the area in a statistical distribution where the null hypothesis is rejected.

**Significance Level**
The significance level, denoted by (\alpha), is the maximum probability of rejecting the null hypothesis when it is actually true. A common value is (0.05).

**p-value**
The p-value shows the probability of obtaining the observed result, or a more extreme result, if the null hypothesis is true. A small p-value provides evidence against the null hypothesis.

**Decision Rule**
A decision rule is used to decide whether to reject or fail to reject the null hypothesis. For example, if (p-value < $\alpha$), the null hypothesis is rejected.

**Data Visualisation**
Data visualization is the use of graphs, charts, and diagrams to present data clearly and support the interpretation of hypothesis testing results.

**Bias**
Bias is an unfair influence in data collection, analysis, or interpretation that may lead to incorrect or misleading conclusions.

**Ethical Use of Data**
Ethical use of data means collecting, analysing, and reporting data honestly, fairly, and responsibly while protecting privacy and avoiding misuse.

**Communicating Results**
Communicating results means presenting findings clearly, accurately, and in relation to the original hypothesis so that readers can understand the conclusion.

# EXERCISE

## Multiple Choice Questions

1. A hypothesis is:
   (a) A final result (b) A test value
   (c) A proposed explanation (d) A dataset
2. The null hypothesis represents:
   (a) An alternative idea (b) No effect or no difference
   (c) A final conclusion (d) A prediction model
3. The symbol used for the null hypothesis is:
   (a) (H_1) (b) (H_2)
   (c) (H_0) (d) (H_x)
4. A p-value is used for:
   (a) Measuring accuracy (b) Testing a hypothesis
   (c) Measuring variance (d) Data storage
5. If (p-value $< \alpha$), the decision is to:
   (a) Accept (H_0) (b) Reject (H_0)
   (c) Ignore data (d) Recalculate the mean
6. The critical region is the:
   (a) Safe area (b) Rejection area
   (c) Data range (d) Sample space
7. An example of a test statistic is:
   (a) Mean (b) t-value
   (c) Mode (d) Median
8. Data visualisation is used to:
   (a) Store data (b) Present results clearly
   (c) Delete data (d) Calculate the mean
9. A chart useful for comparison is:
   (a) Pie chart (b) Bar chart
   (c) Table (d) Text
10. Bias in data means:
    (a) Correct data (b) Balanced data
    (c) Unfair representation (d) Random data

## Short Questions

1. What is a hypothesis?
2. What is a research question?
3. Define null hypothesis ($H_0$).
4. What is a test statistic?
5. What is meant by p-value?

6. What is a critical region?
7. What are the basic steps in hypothesis testing?
8. Why is data visualization important?
9. What is bias in data collection?
10. Why is ethical use of data important?

## Long Questions

1. Explain the concept of hypothesis testing and its role in data analysis.
2. Describe different types of hypotheses and explain how to formulate correct hypotheses.
3. Explain key concepts in hypothesis testing, including test statistics, critical region, and p-value.
4. Discuss the steps involved in performing hypothesis testing with a suitable example.
5. Explain the importance of data visualization in hypothesis testing and how graphs help present results.
6. Discuss ethical issues in data science, including bias, ethical use of data, and responsible communication.
7. Explain how to interpret hypothesis testing results and connect conclusions to hypotheses.
8. Describe how findings should be communicated clearly in data analysis and why it is important.

## Answer Key

1. C
2. B
3. C
4. B
5. B
6. B
7. B
8. B
9. B
10. C

# Chapter 8

# Applications of Computer Science

**Student Learning Outcomes:**

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

- Evaluate how different stakeholders' cultures, values, and sometimes conflicting interests affect AI system designs, and assess policies that protect stakeholders' interests.
- Design innovative applications relevant to Pakistan using emerging technologies such as Artificial Intelligence (AI), Internet of Things (IoT), Cloud Computing, and Blockchain.

## Introduction

In this chapter, students will explore the applications of computer science through emerging technologies. They will learn about Artificial Intelligence (AI) and its use in everyday applications such as voice assistants, chatbots, and image recognition, along with its benefits and challenges. The chapter introduces the Internet of Things (IoT) and how connected devices collect and share data for automation. The chapter highlights the cultural and ethical implications of AI, including fairness, transparency, and the role of different stakeholders. Students will examine how emerging technologies can be applied to solve national challenges in Pakistan, with a focus on innovative solutions like smart agriculture. Finally, they will learn about policies, regulations, and responsible innovation, emphasizing the importance of ethical and sustainable use of technology.

## 8.1 Understanding Emerging Technologies

Emerging technologies refer to advanced innovations that are transforming how people live, work, and communicate. Technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), Cloud Computing, and Blockchain are reshaping industries and creating smarter systems for managing data, automation, and connectivity (as shown in Figure 8.1). These technologies drive innovation across healthcare, education, transportation, and agriculture, making processes faster and more efficient. Understanding emerging technologies is vital in today's digital age. They offer new solutions to global and local challenges but also introduce issues such as data security, ethical concerns, and dependence on automation. By learning

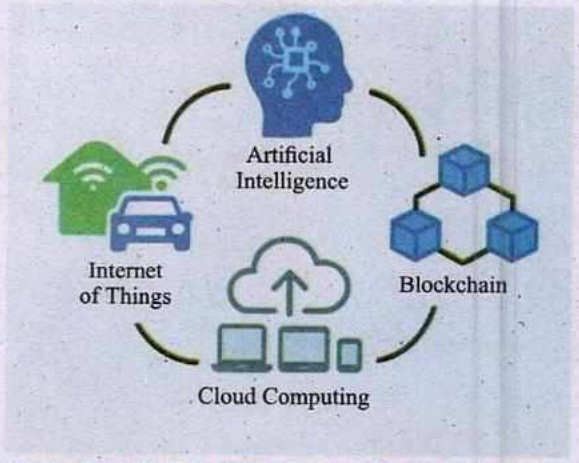



**Figure 8.1: AI, IoT, Cloud Computing, and Blockchain Working Together to Shape the Modern World.**

about these technologies, individuals can better understand their potential, prepare for future developments, and use them to improve society responsibly.

**TIDBIT**

Emerging technologies like AI, IoT, Cloud Computing, and Blockchain are shaping the next digital revolution, connecting people, data, and devices across the world.

## Artificial Intelligence (AI)

Artificial Intelligence (AI) is a branch of computer science that enables machines to perform tasks that typically require human intelligence, (Figure 8.2). It allows computers to analyze data, recognize patterns, and make decisions with minimal human input. AI systems learn from experience through a process called *machine learning*, where algorithms are trained on large sets of data to identify trends and improve performance over time. The more data the system processes, the more accurate its results become.

AI works by combining data, algorithms, and computing power to perform intelligent tasks such as understanding speech, detecting objects in images, and predicting outcomes. It uses technologies like Neural Networks (NN) and Natural Language Processing (NLP) to simulate how the human brain processes information. This allows AI to solve complex problems, automate processes, and assist in areas where traditional programming cannot handle uncertainty or vast amounts of information.

### Common AI Applications (e.g., voice assistants, chatbots, image recognition)

Artificial Intelligence is present in many tools and services people use daily. Voice assistants like Siri, Alexa, and Google Assistant understand spoken commands and respond with useful information or actions. Chatbots on websites help customers by answering questions, while image recognition systems identify faces, objects, or medical conditions from photos and scans. AI is also used in recommendation systems on platforms like YouTube, Draz, and Amazon to suggest videos or products based on user preferences. In transportation, self-driving cars use AI to interpret their surroundings and make safe driving decisions. These applications make technology more interactive, efficient, and adaptive, offering personalized experiences in almost every digital service today.

### Benefits and Challenges of AI in Society.

AI brings significant benefits to modern life by increasing efficiency, accuracy, and convenience. It can perform repetitive tasks faster than humans, assist doctors in diagnosing diseases, and help farmers monitor crops more effectively. For example, DeepMind's AI can analyze 3D eye scans and help predict eye diseases such as glaucoma before clear symptoms become visible to doctors. In business, companies such as Amazon and Daraz use AI chatbots to answer customer questions, such as "Where is my order?", for thousands of users at the same time. Businesses also use AI to analyze customer behavior, predict trends, and improve decision-making. AI supports education through personalized learning and adaptive tutoring systems.

Despite these benefits, AI introduces challenges that must be managed carefully. Concerns include data privacy, job displacement, and bias in algorithms that may lead to unfair outcomes. (as shown in Figure 8.2) Ethical issues also arise when AI systems make decisions that affect people's lives. Balancing innovation with responsibility ensures that AI remains a tool that benefits humanity rather than harms it. Understanding these challenges helps promote fair, transparent, and ethical use of AI in all sectors.

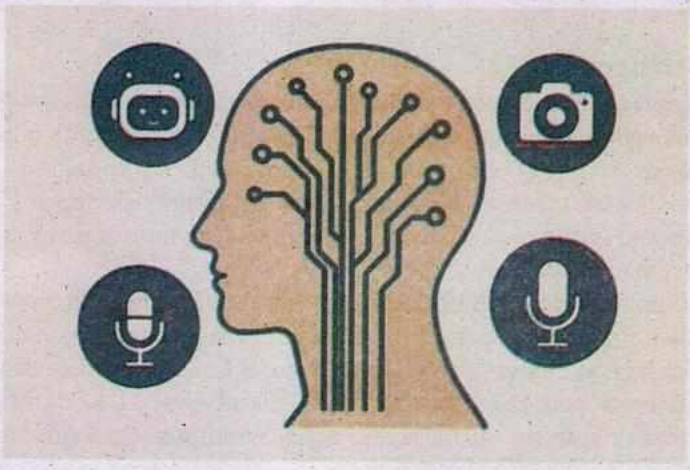


**Figure 8.2: Artificial Intelligence systems mimic human intelligence to perform tasks like vision, speech, and decision-making.**

**TIDBIT**

Farmers use AI with drones and sensors to monitor crops, helping them increase yield and reduce waste.

## Internet of Things (IoT)

The Internet of Things (IoT) refers to a network of physical devices connected through the internet that can collect, share, and act on data. These devices often include sensors, processors, and communication hardware, allowing them to work together intelligently. IoT connects everyday objects such as lights, thermostats, and appliances so they can be monitored and controlled remotely. This interconnection helps improve efficiency, convenience, and real-time decision-making in daily life and industrial operations.

**DO YOU KNOW?**

The IoT connects everyday devices like fridges, cars, and lights so they can "talk" to each other automatically.

IoT systems operate by linking devices through wireless communication technologies like Wi-Fi, Bluetooth, or cellular networks. Each connected device communicates with others or a central platform to exchange information automatically. This continuous interaction creates a smart environment where machines and systems work together with minimal human effort.

In smart homes, IoT enables residents to control lighting, heating, curtains, and security systems through mobile applications or voice commands, as shown in Figure 8.3. For example, smart LED bulbs can be turned on or off through a smartphone, set to turn on automatically at 6:00 p.m. Similarly, smart curtains or blinds can open automatically with the morning alarm and close when the room temperature rises above 35°C to help keep the room cool. A smart thermostat can also adjust temperature based on user preferences, while security cameras can send alerts directly to smartphones. These systems not only add comfort but also help conserve energy and improve safety.

In healthcare, IoT sensors can monitor patient conditions such as heart rate, blood pressure, or glucose levels in real time. This allows doctors to track patient health remotely and respond quickly in emergencies. Similarly, in agriculture, IoT devices measure soil moisture, temperature, and weather patterns to optimize irrigation and crop management, increasing yield and reducing waste.

**TIDBIT**

Smart homes, wearable fitness bands, and smart farms are all powered by IoT technology.

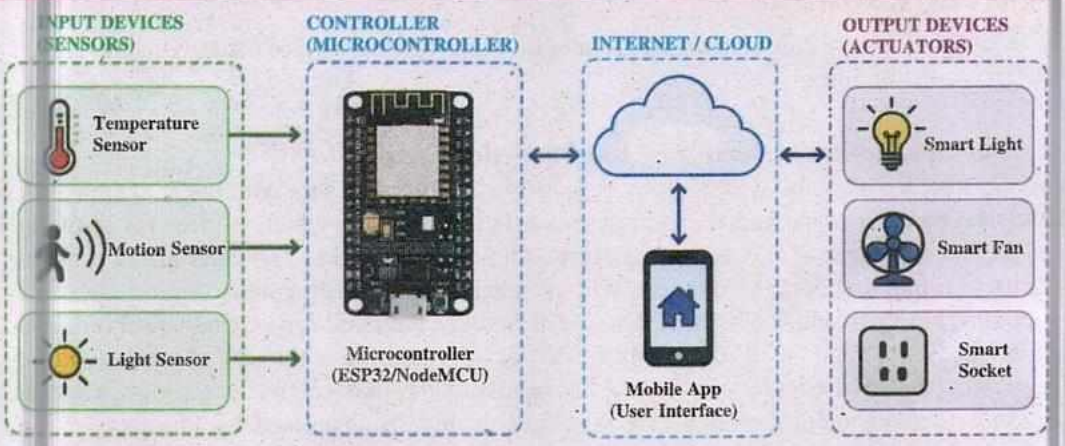



**Figure 8.3: Smart Home using IoT: Connected devices such as lights, thermostat, and security systems can be monitored and controlled remotely through the internet.**

**Data Collection, Communication, and Automation.**

IoT works through three main functions: data collection, communication, and automation. Sensors collect data from the environment, such as temperature, movement, or humidity. This data is then sent to cloud-based systems or servers where it is processed and analyzed. The analyzed data can trigger specific actions automatically, such as turning off lights when no one is in a room or starting irrigation when the soil becomes dry.

Communication between devices ensures that systems remain synchronized and efficient. Automation reduces the need for manual control, saving time and minimizing human error. By combining real-time data with intelligent decision-making, IoT creates smarter, safer, and more efficient environments for homes, industries, and communities.

**CLASS ACTIVITY**

Sketch a smart home system that uses IoT devices (lights, fans, sensors). Label how each device collects and communicates data.

## Cloud Computing

Cloud computing is a technology that allows users to store, manage, and process data on remote servers instead of local computers (as shown in Figure 8.4). These servers are hosted on the internet and managed by cloud service providers. This means users can access their data and applications anytime, from anywhere, using an internet connection. The cloud eliminates the need for powerful personal computers, as most of the processing is done remotely.

By using cloud computing, individuals and businesses can use online services such as data storage, web hosting, and software applications without having to maintain physical hardware. It supports modern digital tools like streaming platforms, online learning systems, and collaborative workspaces. This technology forms the backbone of many services people use daily, making computing more flexible and connected.

**DO YOU KNOW?**

When you save a document on Google Drive or iCloud, you're actually using Cloud Computing.

**Advantages: Scalability, Accessibility, and Cost-effectiveness.**

One of the key advantages of cloud computing is scalability. Users can easily increase or decrease their storage space and computing power based on need, which is ideal for growing businesses and organizations. Accessibility is another benefit, as cloud services allow users to work from multiple devices and locations without transferring files manually.

Cloud computing is also cost-effective because it reduces the need for expensive servers and maintenance staff. Users usually pay only for the services they use through a pay-as-you-go model, which helps lower operational costs. These advantages make cloud technology practical for both individuals and companies that require reliable and efficient digital solutions.

**Examples: AWS, Microsoft Azure.**

Many popular platforms use cloud computing to provide everyday digital services. Amazon Web Services (AWS) offers a wide range of cloud-based services for web hosting, data storage, and application development, widely used by businesses around the world. Microsoft Azure provides similar capabilities, enabling developers to create and manage applications through the cloud (as shown in Figure 8.4).

These platforms show how cloud computing supports various activities, from personal file storage to large-scale enterprise solutions. Whether it is saving documents, hosting websites, or running software remotely, cloud computing has become an essential part of modern digital infrastructure.

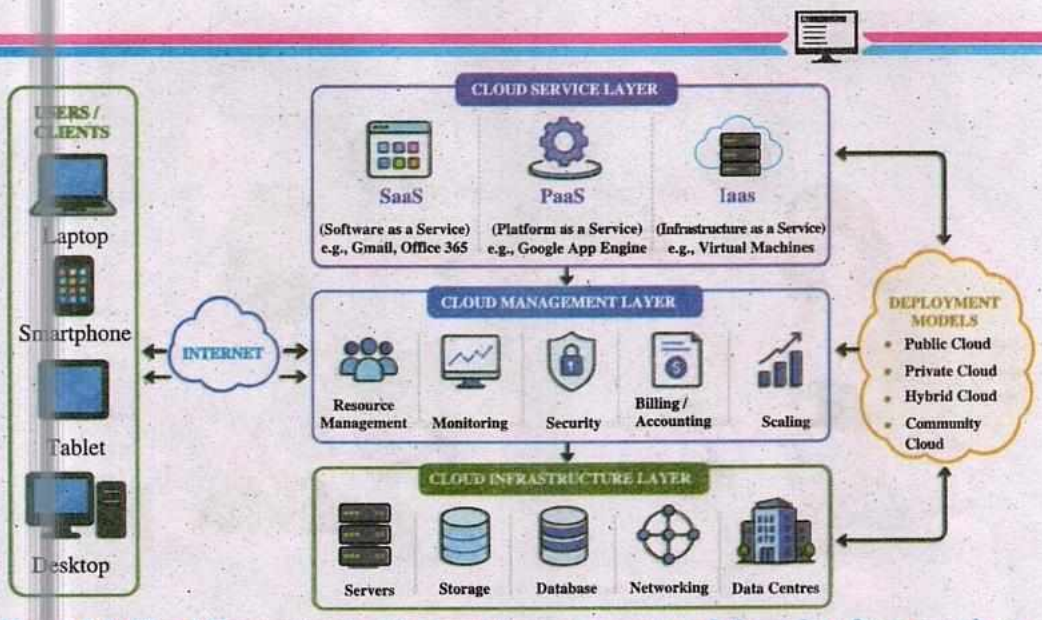



**Figure 8.4: Cloud Computing enables users to store, access, and share data from anywhere through the internet.**

## Blockchain Technology

Blockchain technology is a digital system that securely stores and shares information across many computers instead of a single central server. Each verified record, or *block*, links to the previous one, forming an unchangeable chain. Because every computer holds a copy, altering data without detection is nearly impossible. This decentralized structure increases transparency, security, and trust by removing intermediaries like banks or governments. It allows direct, accurate, and accountable exchanges between individuals and organizations.

Blockchain has wide applications across industries. In finance, it powers cryptocurrencies such as Bitcoin for secure, peer-to-peer transactions. In supply chain management, it tracks goods from origin to customer, ensuring authenticity. In data security, it provides tamper-proof storage for records like legal documents, medical files, and certificates. Governments, companies, and schools use blockchain to manage land records, supply chains, and academic credentials. Overall, blockchain enhances reliability, transparency, and efficiency in digital systems.

**TIDBIT**

From cryptocurrency to supply chain tracking, blockchain builds trust through transparency.

## 8.2 Cultural and Ethical Implications of AI Systems

Artificial Intelligence (AI) influences how people live, work, and interact with technology, making it essential to consider its cultural and ethical effects. AI systems are not only technical creations but also social tools that impact individuals, communities, and governments. (as shown in Figure 8.5).

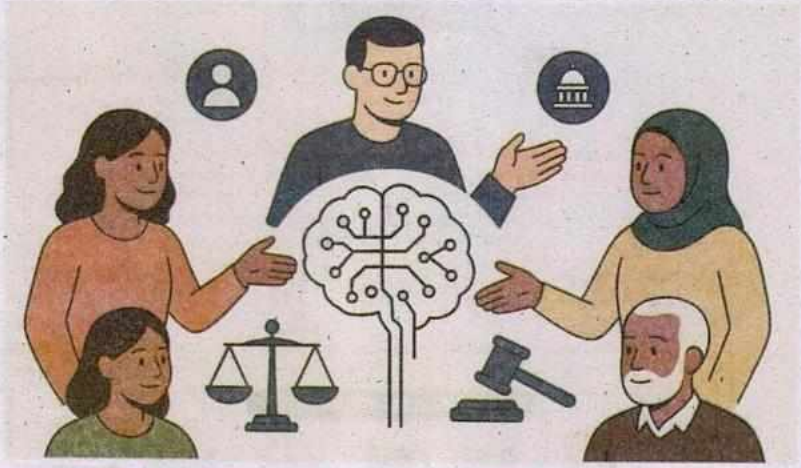


Figure 8.5: Ethical AI Design Balances the Interests of Different Stakeholders While Ensuring Fairness and Inclusivity.

**CLASS ACTIVITY**

Debate: "Should AI make decisions that affect people's lives, such as job hiring or grading students?"

## Stakeholders in AI Systems

AI systems involve multiple stakeholders, each playing a key role in their creation and use. Developers design and program AI models, users interact with them in daily activities, governments regulate their applications, and communities experience their impact. Each group contributes to the overall ecosystem of AI and influences how it evolves.

**Developers:** Developers design and build AI systems focusing on functionality, innovation, and performance. They ensure systems are accurate, fair, and ethically developed.

**Users:** Users interact with AI systems and expect them to be easy to use and reliable. They are concerned about privacy, security, and correct outcomes.

**Governments:** Governments create policies and regulations to ensure safe and fair use of AI. They work to protect public interests and enforce legal compliance.

**Communities:** Communities focus on preserving social values and cultural norms. They promote responsible and ethical use of AI in society.

## Cultural Awareness and Diversity

Cultural diversity affects how AI systems are designed, perceived, and implemented around the world. Cultural values influence user expectations regarding privacy, communication styles, and technology adoption. Designers must consider local traditions and behaviors to make AI tools accessible and respectful to different cultures.

For instance, privacy preferences vary between regions; some societies are comfortable with data sharing, while others demand strict confidentiality. Language models must include diverse languages and dialects to avoid bias toward dominant cultures. Similarly, AI-driven decisions should reflect sensitivity to local customs and social norms to ensure fairness and acceptance.

**DO YOU KNOW?**

An AI that understands different languages and cultures can serve people from all backgrounds more effectively.

## Ethical Principles in AI

**Fairness, Accountability, Transparency, and Safety in AI Systems.**

Ethical AI development requires fairness to prevent discrimination, accountability to assign responsibility, transparency to explain how systems make decisions, and safety to protect users from harm. For example, if an AI system is used for student admission, it should not unfairly favor or reject students based on gender, region, language, or background. The university must also be able to explain why a student was selected or rejected and who is responsible if the system makes an unfair decision. Similarly, in healthcare, an AI system used for disease diagnosis must be tested carefully so that it does not give harmful or misleading results. These principles ensure that AI remains a tool for human benefit and does not create unintended negative consequences.

**Avoiding Bias and Ensuring Inclusivity in AI Design.**

AI systems often learn from large datasets, which may contain hidden biases. For example, if a hiring system is trained mostly on past records where men were selected more often than women, the AI may unfairly prefer male candidates. Similarly, if a face recognition system is trained mainly on images of one age group, skin tone, or region, it may perform poorly for people from other groups. Developers must carefully review and adjust these datasets to avoid unfair treatment of individuals or groups. Inclusivity in AI design means considering diverse perspectives and ensuring that technology serves all people equally, regardless of gender, ethnicity, or background.

**Conflict of Interests Among Stakeholders**

Conflicts often arise when stakeholders have competing priorities. Businesses aim to collect data for innovation and profit, while users value privacy and control over personal information. Governments must balance economic growth with protecting citizens' rights and maintaining ethical standards.

A common example of conflict is data collection practices. Companies may gather large amounts of user data to improve AI performance, but without explicit consent, this violates privacy and trust. Transparent data policies and user-friendly consent mechanisms help achieve a fair balance between functionality and ethics.

**TIDBIT**

Balancing privacy and innovation is one of the biggest challenges in today's tech-driven world.

**CLASS ACTIVITY**

Role-play a company meeting where developers want to collect user data and privacy advocates oppose it. Discuss solutions.

## Policy and Legal Frameworks

Several international frameworks guide responsible AI use. The General Data Protection Regulation (GDPR) from the European Union enforces strict rules on data privacy and user rights. AI Ethics Guidelines developed by organizations worldwide emphasize accountability, fairness, and respect for human dignity. These frameworks encourage nations and companies to adopt responsible AI practices.

For Pakistan, creating national AI policies is crucial to promote innovation while ensuring ethical compliance. Such policies can regulate data protection, encourage research, and support the development of AI solutions that align with local values and needs. Implementing ethical standards helps build public trust, attract global partnerships, and ensure that AI contributes positively to national growth.

# 8.3 Designing Applications for Pakistan Using Emerging Technologies

Emerging technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), Cloud Computing, and Blockchain offer new opportunities for solving Pakistan's pressing challenges (as shown in Figure 8.6). By combining innovation with local understanding, students can design applications that improve lives, enhance productivity, and support sustainable development. These technologies can transform sectors like healthcare, education, agriculture, and governance when used responsibly and creatively.

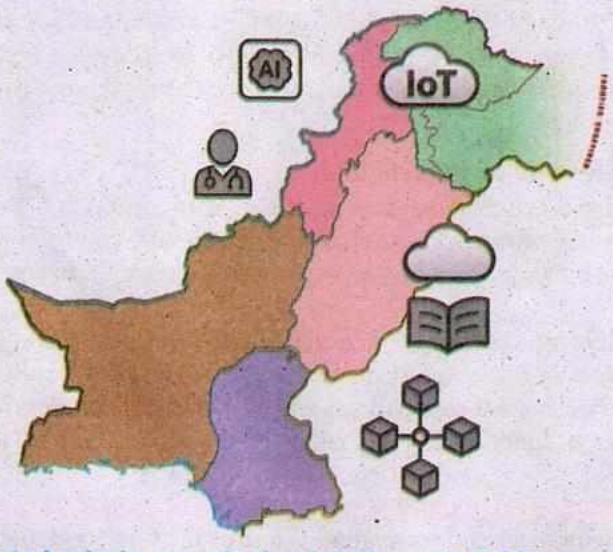



**Figure 8.6: Emerging technologies can solve local challenges in Pakistan across key sectors like agriculture, health, and governance.**

## Identifying National Challenges

Pakistan faces several national challenges that can be addressed through technology. The healthcare sector struggles with limited accessibility and resource shortages; education faces barriers in quality and outreach; agriculture deals with inefficiency and low productivity; and



Sig8 SideB Process Black 07/07/2026 26 Computer 12 pecta ishaq

governance requires transparency and efficiency. Recognizing these issues is the first step toward developing meaningful solutions.

Emerging technologies can bridge many of these gaps. AI can analyze medical data for faster diagnoses, IoT can monitor environmental conditions in agriculture, Cloud Computing can store and distribute educational content efficiently, and Blockchain can ensure transparency in governance. These innovations can make public services smarter, more accessible, and more reliable for citizens.

**DO YOU KNOW?**

Technology can help address major challenges in Pakistan, such as water shortages, limited access to education, and healthcare delivery.

## Innovative Application Design

Designing applications using emerging technologies begins with identifying a real-world problem and selecting the right combination of tools to solve it. For example, AI can be used for intelligent tutoring systems, IoT for smart city management, Cloud for secure data access, and Blockchain for record authentication. Students are encouraged to brainstorm ideas that directly address Pakistan's social and economic needs.

Innovation thrives when creativity meets purpose. Students can use design-thinking methods to develop user-focused, practical solutions tailored to local conditions. Projects that address issues such as water management, traffic control, or education accessibility help foster critical thinking and contribute positively to community development.

**CLASS ACTIVITY**

Sketch a simple app interface (on paper or digitally) that uses AI, IoT, or Cloud to solve a real-world problem.

## Technology Integration

Effective solutions often integrate more than one emerging technology. Combining AI's intelligence, IoT's connectivity, Cloud's scalability, and Blockchain's security can create powerful, data-driven systems. This integration leads to smarter applications that not only solve problems but also adapt and improve over time.

A smart irrigation system can use IoT sensors to measure soil moisture, AI algorithms to predict crop water needs, and Cloud storage to collect and analyze data. This helps farmers optimize water use, reduce costs, and increase yields. Such integrated systems demonstrate how technology collaboration can support sustainable agriculture and environmental conservation in Pakistan.

**CLASS ACTIVITY**

Form mixed groups (technology, social science, business). Design a collaborative project that uses technology to solve a community issue.

## Ethical and Cultural Considerations

When designing technology-based solutions, it is essential to consider Pakistan's diverse cultures, traditions, and values. Applications must be inclusive, accessible, and respectful of

local customs. For instance, AI systems should use language and content that reflect regional diversity and social sensitivities.

Technological innovation must always prioritize ethical responsibility. Misuse of data, invasion of privacy, or culturally inappropriate content can create distrust and harm communities. Students should design applications that promote fairness, respect, and positive social impact , ensuring that technology remains a positive force for progress.

**TIDBIT**

Technology succeeds when it respects local traditions, beliefs, and user needs.

### Case Study: Smart Agriculture in Pakistan

**Using IoT Sensors for Soil Monitoring**

IoT sensors can collect data about soil moisture, temperature, and nutrients in real time. This information helps farmers make data-driven decisions, such as when and how much to water their crops, improving efficiency and conserving resources.

**AI for Predictive Weather Analysis**

AI systems can analyze weather data to predict rainfall, drought, or temperature patterns. These predictions help farmers plan their activities and protect crops from climate-related damage, leading to better yields and reduced losses.

**Cloud for Data Storage and Blockchain for Supply Chain Transparency.**

Cloud Computing provides scalable storage and quick access to agricultural data for farmers, researchers, and policymakers. Blockchain adds transparency to the supply chain by recording each stage of crop production and distribution, reducing fraud and ensuring fair trade. Together, these technologies create a sustainable and trustworthy agricultural ecosystem for Pakistan's future.

## 8.4 Policies, Regulations, and Responsible Innovation

As emerging technologies continue to advance, clear policies and responsible innovation practices are essential to ensure their safe and ethical use (as shown in Figure 8.7). Laws, governance frameworks, and collaboration across different disciplines help balance progress with accountability. By understanding how policies shape technological growth, students can learn to design solutions that uphold fairness, inclusivity, and contribute to social good.

**DO YOU KNOW?**

Combining AI, IoT, Cloud, and Blockchain can create smarter and more efficient systems for everyday life.

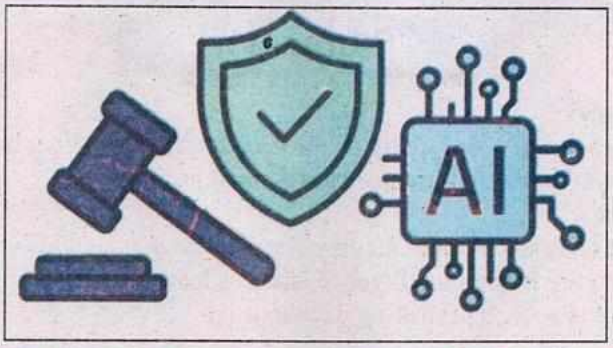



**Figure 8.7: Policies and Ethical Guidelines Ensure Responsible Innovation and Protect User Rights in Technology Development.**

## Understanding AI Governance

Emerging technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), and Blockchain affect many aspects of human life, from privacy and employment to decision-making and security. Without proper laws and regulations, these technologies can be misused or create unfair outcomes. Policies help define how data should be collected, stored, and used, ensuring safety and protecting individuals' rights.

Globally, frameworks like the General Data Protection Regulation (GDPR) in Europe set standards for data protection and privacy. International AI ethics guidelines also promote transparency and fairness in automated decision-making. At the national level, countries are developing their own AI strategies to guide innovation responsibly. For Pakistan, adopting similar frameworks can ensure that technological growth aligns with ethical standards and safeguards citizens' interests.

## Developing Responsible Technology

Responsible innovation means focusing on human welfare rather than only financial gains. Technologies should be developed to improve quality of life, create opportunities, and minimize harm. When designing systems, developers must consider the long-term social and environmental impacts, ensuring that technology benefits communities equitably.

**TIDBIT**

The best technologies put people first, ethics and empathy are key ingredients in innovation.

## Collaboration Across Disciplines

Building responsible and fair technology requires collaboration among professionals from various fields. Legal experts ensure compliance with regulations, sociologists study the social impact of technology, and ethicists guide moral decision-making. Working together, these experts help create balanced solutions that protect people while supporting innovation.

Technological solutions should represent the diversity of the populations they serve. By considering differences in language, culture, and accessibility, developers can design systems that work for everyone, not just a selected group. Interdisciplinary collaboration ensures that innovation remains inclusive, fair, and sustainable, paving the way for technology that truly benefits society.

## Summary

**Emerging Technologies**
Emerging technologies are new or rapidly developing technologies that have the potential to change how people live, work, communicate, and solve problems.

**Artificial Intelligence**
Artificial Intelligence (AI) is a branch of computer science that enables machines to perform tasks that normally require human intelligence, such as learning, reasoning, recognising images, understanding speech, and making decisions.

**Internet of Things**
The Internet of Things (IoT) refers to physical devices connected to the Internet that can collect, share, and act on data. Examples include smart watches, smart lights, smart cameras, and soil-monitoring sensors.

**Cloud Computing**
Cloud computing is the use of Internet-based services to store data, run applications, and access computing resources without depending only on a local computer or server.

**Blockchain**
Blockchain is a digital record-keeping technology in which data is stored in linked blocks. It is commonly used to support secure, transparent, and tamper-resistant transactions.

**Stakeholders**
Stakeholders are individuals or groups affected by a technology system. In AI systems, stakeholders may include users, developers, organisations, governments, and society.

**Ethical AI**
Ethical AI means designing and using AI systems in a fair, transparent, accountable, safe, and responsible manner.

**Bias in AI**
Bias in AI occurs when an AI system gives unfair results because of biased data, poor design, or unequal representation of different groups.

**Inclusivity**
Inclusivity means designing technology so that it can serve people from different genders, cultures, languages, regions, and backgrounds fairly.

**Conflict of Interest**
A conflict of interest occurs when the goals or benefits of one stakeholder may negatively affect the rights, safety, or interests of others.

**AI Governance**
AI governance refers to the rules, policies, and guidelines used to ensure that AI systems are developed and used responsibly.

**Responsible Innovation**
Responsible innovation means creating technology that solves real problems while considering ethics, safety, culture, fairness, and long-term social impact.

**Smart Agriculture**
Smart agriculture is the use of technologies such as IoT, AI, cloud computing, and blockchain to improve farming.

**Technology Integration**

Technology integration means combining different technologies to build a more useful solution. For example, an agriculture system may use IoT for data collection, AI for prediction, cloud computing for storage, and blockchain for secure records.

**Policy and Legal Frameworks**

Policy and legal frameworks are rules and laws that guide how technologies should be developed, used, and controlled to protect individuals, organisations, and society.

# EXERCISE

## Multiple Choice Questions

1. Artificial Intelligence (AI) is best described as:
   (a) A system that connects devices to the Internet
   (b) A program that performs human-like tasks
   (c) A method of storing files online
   (d) A social media algorithm
2. The Internet of Things (IoT) is mainly used to:
   (a) Allow devices to communicate automatically
   (b) Store documents securely
   (c) Manage Internet connections
   (d) Create digital art
3. An example of cloud computing is:
   (a) Google Drive
   (b) Microsoft Word
   (c) Smartwatch
   (d) USB flash drive
4. Blockchain technology is known for:
   (a) Decentralized and secure records
   (b) Temporary file storage
   (c) High-speed networking
   (d) Real-time video editing
5. An ethical issue in AI systems is:
   (a) Data bias in decision-making
   (b) Using colorful interfaces
   (c) Quick data analysis
   (d) Saving battery power
6. The purpose of AI governance policies is to:
   (a) Increase automation speed
   (b) Protect privacy and ensure responsible use
   (c) Reduce programming errors
   (d) Promote entertainment content
7. In agriculture, IoT and AI can work together to:
   (a) Automate soil and crop monitoring
   (b) Create social media advertisements
   (c) Manage financial transactions
   (d) Generate artwork
8. Transparency in agriculture supply chains can be improved using:
   (a) Blockchain and cloud computing
   (b) HTML and CSS
   (c) GPS and Wi-Fi
   (d) Python and Excel
9. Responsible technology development should:
   (a) Focus only on business profit
   (b) Ensure fairness and accountability
   (c) Avoid community input
   (d) Limit global access

10. Cultural awareness in AI design ensures:
    (a) Faster algorithm performance
    (b) Respect for local values and norms
    (c) Reduced data collection
    (d) Larger system databases

## Short Questions

1. What is Artificial Intelligence (AI) and how is it used in daily life?
2. How does the Internet of Things (IoT) connect smart devices together?
3. What are the main advantages of using Cloud Computing?
4. How does Blockchain ensure security and transparency in transactions?
5. Why is it important to consider ethics while developing AI systems?
6. What role do stakeholders play in the design and regulation of AI technologies?
7. How can emerging technologies help solve national challenges in Pakistan?
8. What is the importance of cultural awareness when designing technological solutions?
9. How can AI, IoT, and Cloud work together to improve agriculture?
10. Why are laws and policies necessary for the responsible and safe use of technology?

## Long Questions

1. Explain how Artificial Intelligence (AI), the Internet of Things (IoT), Cloud Computing, and Blockchain are transforming modern industries. Give suitable examples for each technology.
2. Discuss the benefits and challenges of Artificial Intelligence (AI) in society. How can ethical principles ensure fair and responsible use of AI?
3. Describe how IoT devices collect and communicate data for automation. How is IoT improving sectors such as healthcare, agriculture, and smart homes?
4. Explain how Cloud Computing supports scalability, accessibility, and collaboration. Compare different cloud platforms like Google Drive, AWS, and Microsoft Azure.
5. What is Blockchain technology? Discuss how its decentralized nature improves transparency and security in financial and non-financial applications.
6. Identify key national challenges in Pakistan and describe how emerging technologies (AI, IoT, Cloud, Blockchain) can provide effective and innovative solutions.
7. Discuss the ethical and cultural considerations involved in designing AI systems. How can developers ensure inclusivity and respect for local values?
8. Describe the importance of policies, regulations, and interdisciplinary collaboration in promoting responsible innovation. How can Pakistan benefit from adopting global AI governance standards?

## Answer Key

| | | | |
|---|---|---|---|
| 1. | B | 6. | B |
| 2. | A | 7. | A |
| 3. | A | 8. | A |
| 4. | A | 9. | B |
| 5. | A | 10. | B |

# Chapter 9

# Cybersecurity and Safe Digital Collaboration

**Student Learning Outcomes:**

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

- Identify and apply safe practices when collaborating on digital or online platforms.
- Understand of common types of cybersecurity problems and topics such as 2FA, biometric verification, and secure techniques for transmitting data.
- Collaborate on creation of digital applications and understand strategies to design for equity and equal access to information.
- Explore digital entrepreneurship concepts and develop simple technology-based business ideas using digital platforms and collaborative tools.

## Introduction

In this chapter, students will explore how technology supports safe, secure, and responsible participation in the digital world. They will learn about online safety, cybersecurity, data protection, privacy settings, and secure collaboration using modern digital platforms. The chapter also introduces digital entrepreneurship, innovation, and the skills needed to identify business opportunities and solve real-world problems through technology. Students will explore ethical, legal, social, and environmental aspects of computing, including intellectual property, data privacy, cybersecurity laws, and sustainable computing practices. They will also learn the importance of teamwork, accessibility, digital communication, and responsible use of technology in personal, educational, and professional settings (as shown in Figure 9.1). By the end, students will understand how to use technology safely, responsibly, ethically, effectively, and creatively in daily life and future careers.

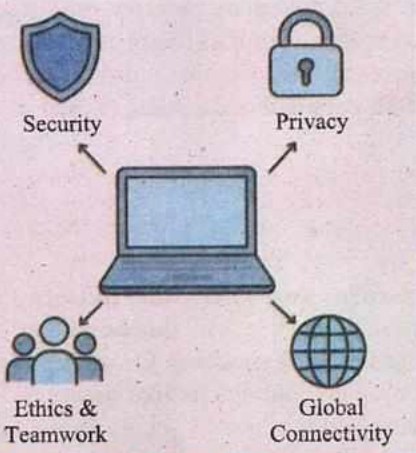



**Figure 9.1: Cybersecurity, Ethics, and Digital Collaboration in Today's Connected World**

## 9.1 Safe and Responsible Digital Practices

This section explains how to use technology wisely and protect personal information while working online. Students learn the importance of safe habits, like using strong passwords, keeping software updated, and avoiding unsafe websites (as shown in Figure 9.2). Practicing responsible digital behavior helps prevent data loss, hacking, and identity theft. By following safety rules, users can enjoy the benefits of technology without unnecessary risks. These habits build trust and create a secure digital environment for everyone.

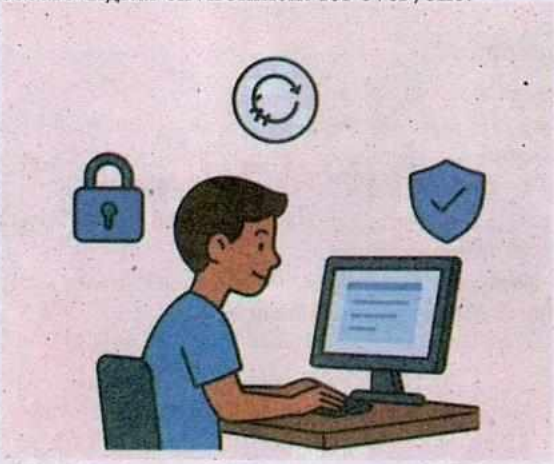


Figure 9.2: Safe Digital Practices

**DO YOU KNOW?**

A strong password with 12 characters can take millions of years for a computer to crack, but a weak one can be cracked in seconds.

### Importance of Online Safety

Online safety means protecting yourself and your information while using the internet. It includes being careful about what you share, where you click, and how you store your data. Students learn that online actions leave digital footprints that can be tracked. By understanding the importance of safety, users can avoid scams, viruses, and cyber threats. Staying alert and cautious online helps keep both personal and school_related data secure.

**TIDBIT**

Understanding threats and preventive measures, reduces the chance of cybercrime.

#### Understanding Strong Passwords and Their Role in Data Protection

A strong password is a combination of letters, numbers, and symbols that helps protect your accounts from hackers. Using unique passwords for different accounts reduces the chance of being hacked. It's best to avoid personal details like names or birthdays. Passwords should be changed regularly and not shared with others. A secure password acts as the first line of defense in protecting important files and data online.

### Risks of Downloading Suspicious Software and Unsafe Browsing Habits

Downloading programs or files from unknown websites can harm your computer or steal your information. Some downloads hide viruses or spyware that collect data without permission. Unsafe browsing, such as visiting fake websites or clicking unknown links, can also lead to scams or identity theft. It is important to check sources before downloading and use antivirus tools for extra protection. Safe browsing habits help keep both devices and data secure.

## Software Maintenance and Updates

Software maintenance means keeping programs and applications in good working condition by updating them regularly. Updates often fix errors, improve performance, and add new features that make software safer to use. Students learn that software companies release updates to close security gaps that hackers might exploit. Ignoring updates can lead to slow systems, crashes, or data theft. Regular maintenance keeps computers reliable, efficient, and protected from cyber threats.

When software is not updated, it becomes an easy target for hackers. Older versions may have security holes that allow unauthorized access to files or personal data. Outdated software can also cause compatibility issues with new systems or tools. Using such programs increases the risk of malware infections and data breaches. Regularly updating software reduces these risks and keeps your system safe and dependable.

**CLASS ACTIVITY**

Students compare features of updated vs outdated software and discuss how old versions can be attacked.

## Privacy and Security Settings

Privacy and security settings help users control how their personal information is used and shared online. These settings are found in most apps, websites, and digital platforms. Students learn how to adjust them to protect personal data such as location, contacts, and browsing history. Managing privacy helps prevent misuse of information by unwanted sources or advertisers. Understanding and using these settings keeps online activities more private and secure. For example, on Instagram, users can make their account private so that only approved followers can see their posts. They can also turn off location sharing to protect their privacy and enable two-factor authentication for extra account security.

### Reading and Managing Privacy Policies

Privacy policies explain how a website or app collects, stores, and uses your information. It is important to read them carefully before creating an account or agreeing to terms. By reviewing these policies, users can decide whether to share personal data or not. Students learn to check what kind of data is being collected and for what purpose. Managing privacy responsibly ensures that personal details are not shared without permission. A privacy policy tells users how their data is collected, used, stored, and shared. For example, when signing up for any social platform, the privacy policy explains what type of data may be collected, such as name, email address, location, browsing activity, and app usage. It may also explain whether this data is shared with advertisers or third-party services.

**TIDBIT**

Regularly reviewing settings ensures protection as platforms update their features.

### Customizing Privacy and Security Settings for Online Platforms

Most online platforms allow users to change their privacy and security settings based on their comfort level. These settings can control who sees your posts, who can contact you, and how data is used. Turning on features like two-factor authentication and limiting app permissions increases safety. The user should regularly review and update their settings as platforms often change their privacy rules. Customizing these options gives users more control over their digital presence and protection. For example, on WhatsApp, users can choose who can see their profile photo, such as everyone, only contacts, or nobody. They can also disable read receipts, commonly known as blue ticks, so that others cannot see whether their messages have been read.

## Safe Online Collaboration

Online collaboration allows people to work together using digital tools from any location. It is important to use secure platforms and follow safety rules when sharing files or communicating online. Students learn to work responsibly in virtual groups while protecting personal and school data. Safe collaboration builds trust and ensures productive and respectful teamwork. For example, while using Google Docs for group work, students should share the document only with trusted classmates. They should also set the permission to "View only" instead of "Edit" if others only need to read the document. This helps protect the document from unwanted changes and keeps the work secure.

### Identifying Secure Collaboration Platforms (e.g., Google Workspace, MS Teams)

Secure collaboration platforms such as Google Workspace and Microsoft Teams provide protected environments for group work. These platforms use features like encrypted communication and access controls to safeguard data. Students can share documents, attend meetings, and communicate safely. Choosing trusted and approved platforms helps protect information and reduces cyber risks. For example, students can use Google Workspace for school projects by sharing a project file only with selected classmates through restricted access. They can also use Google Meet for secure online classes, where only authorized participants are allowed to join. These features help students collaborate safely while protecting their schoolwork and personal information.

### Practicing Digital Etiquette and Responsible Communication

Digital etiquette means communicating politely, respectfully, and clearly in online environments. It includes using appropriate language, respecting others' opinions, and avoiding misuse of communication tools. Students should not share private information or behave disrespectfully during online interactions. Responsible communication promotes teamwork and helps maintain a positive and professional digital environment. For example, in a class group chat, students should respect others' opinions during discussions. They should avoid sending spam messages, unrelated memes, or jokes during serious academic discussions. This helps keep the group useful, respectful, and focused on learning.

# 9.2 Cybersecurity Awareness and Threat Management

Cybersecurity awareness means understanding how to protect computers, networks, and data from attacks. As people rely more on technology, cyber threats have become common in daily

life. Hackers use tricks and harmful software to steal information or damage systems. Being aware of these dangers helps users stay alert and act safely online. Knowing how to detect, prevent, and respond to threats keeps personal and organizational data secure. For example, users should install antivirus software to protect their computers from malware and other security threats. They should also avoid using public Wi-Fi for online banking or other sensitive work because public networks may not be secure.

**TIDBIT**

Avoid offensive language, spam, and sharing confidential information.

## Understanding Cyber Threats

Cyber threats are actions or programs that target computers and networks to cause harm. These include stealing information, damaging systems, or interrupting online services. Understanding how threats work helps users recognize early warning signs. Most attacks can be avoided by safe browsing, using antivirus software, and being cautious with emails and links. Recognizing threats is the first step toward protecting digital systems.

**DO YOU KNOW?**

Phishing is among the most common attacks.

### Common Types of Cyberattacks

There are different kinds of cyber threats. A **virus** spreads through infected files and can damage or delete data. **Phishing** tries to trick people into sharing passwords or personal details through fake emails or websites. **Ransomware** locks data and demands payment to unlock it, while spyware secretly collects user information. **Distributed Denial of Service (DDoS)** attacks overload websites, making them crash.

### How Cyberattacks Affect Individuals and Organizations

Cyberattacks can cause serious damage to both individuals and organizations. People may lose personal files, passwords, or money, while companies may face data breaches and loss of customer trust. Attacks can stop business operations, harm reputations, and lead to financial losses. Even schools and hospitals can be affected by cyber incidents.

## Security Tools and Techniques

Security tools and techniques help protect computers and networks from online threats. They make it harder for hackers or malicious software to access important information. These tools include authentication systems, firewalls, and antivirus programs that check and block harmful activities. Using them properly keeps personal data safe and reduces the chances of cyberattacks. Every computer user should understand and apply these tools to maintain digital safety.

**Two-Factor Authentication (2FA)**

Two-Factor Authentication (2FA) is a security method that requires two forms of verification before allowing access to an account. For example, after entering a password, the user may receive a code on their phone or email to confirm their identity. This extra step makes it much harder for hackers to break in.

**Biometric Verification**

Biometric verification uses physical features such as fingerprints, face scans, or voice patterns to verify identity. Biometrics add strong layers of protection to keep accounts and personal data safe.

**CLASS ACTIVITY**

Students simulate logging into an account with a second step (code from teacher) to understand how 2FA works.

**Use of Firewalls, Intrusion Detection Systems (Ids), and Antivirus Tools**

A firewall acts as a barrier between a computer and the internet, allowing safe data to pass while blocking harmful traffic. Intrusion Detection Systems (IDS) monitor networks for suspicious activities and alert users about possible threats. Antivirus software scans files and removes viruses or malware before they cause damage. Together, these tools form a powerful defense against cyberattacks. Using them regularly ensures that computers and networks remain protected and run smoothly.

## Data Protection and Cryptography

Data protection means keeping digital information safe from unauthorized access, loss, or damage. It ensures that only the right people can view or use the data.

**TIDBIT**

Secure storage and transmission prevent unauthorized access and breaches.

Cryptography is the science of securing information so that only authorized users can understand it. It consists of two core concepts such as encryption and decryption. Encryption changes readable data into coded text using a specific key, while decryption converts it back into its original form when needed. For example, when sending a message online, encryption makes it unreadable to anyone who tries to intercept it. This process ensures that information remains private during transmission. Cryptography is widely used in secure websites, online banking, and messaging apps.

**Secure Methods for Transmitting and Storing Data**

Data must be protected not only when stored on a computer but also when sent over the internet.

Secure methods such as encrypted connections (like HTTPS) keep information safe during transmission. Cloud storage services also use encryption to prevent unauthorized access to saved files. Regular backups help protect data from loss caused by system failure or attacks. Following secure storage and transfer methods ensures that important data remains safe and reliable.

### Detecting and Preventing Intrusions

Detecting and preventing intrusions means identifying and stopping unauthorized access to a computer or network. Cyber intrusions can occur when hackers exploit weaknesses to steal data or damage systems. Monitoring tools and security software help track unusual activity that might indicate an attack. Students learn how early detection prevents major losses and keeps systems running safely. Staying alert and updating security measures are key steps to maintaining protection against intrusions.

**CLASS ACTIVITY**

Students look at sample system logs or screenshots to find suspicious actions.

#### Identifying Vulnerabilities and Suspicious Activities

Vulnerabilities are weaknesses in software, hardware, or user behavior that hackers can exploit. Suspicious activities may include unexpected system slowdowns, unknown programs running, or strange network traffic. Regularly checking system logs and using scanning tools can help find these warning signs. Recognizing patterns of unusual behavior allows users to take quick action. Identifying and fixing vulnerabilities early reduces the chance of successful attacks.

#### Applying Troubleshooting and Preventive Measures

When a security issue is found, troubleshooting helps determine its cause and how to fix it. Preventive measures include installing firewalls, antiviruses and anti-malware, updating software, changing passwords, or removing harmful files. In some cases, restoring data from backups may be necessary. Documenting the problem and the steps taken helps prevent it from happening again.

## 9.3 Digital Collaboration and Equity in Computing

Digital collaboration means using technology to work together on shared tasks, projects, or ideas. It allows people to connect and contribute from different places using the internet. Students learn how teamwork, communication, and responsibility help achieve goals more efficiently in digital spaces (as shown in Figure 9.3).

Equity in computing ensures that everyone, regardless of background or ability, can participate and access digital resources equally. Combining collaboration with fairness helps create inclusive and effective learning and working environments.



Figure 9.3: Digital Collaboration

## Working Together Online

Working together online allows teams to share ideas, exchange feedback, and complete tasks without being in the same location. It promotes cooperation and creativity through shared access to documents, discussions, and resources. Students understand that clear communication and mutual respect are key to successful online teamwork. Learning to collaborate digitally prepares them for modern workplaces where remote work is common. Online teamwork encourages problem solving and builds essential interpersonal skills.

**DO YOU KNOW?**

Teams that communicate clearly finish tasks faster and make fewer mistakes

### Importance of Teamwork and Communication in Digital Environments

Teamwork in digital environments requires clear communication and trust among team members. When everyone contributes ideas and listens to others, projects run smoothly. Effective communication prevents misunderstandings and ensures that all members stay informed about progress. Tools like group chats, emails, and video calls help teams stay connected. Good teamwork encourages creativity, cooperation, and shared responsibility in achieving goals.

### Tools and Platforms for Real-Time Collaboration

Real-time collaboration tools allow multiple users to work on the same project at the same time. Platforms such as Google Workspace, Microsoft Teams, and Zoom make sharing documents, chatting, and meeting online easier. These tools help teams organize tasks, edit files together, and stay updated instantly. Real-time collaboration saves time and reduces errors by keeping everyone aligned. Using these platforms efficiently helps teams work faster and more effectively in digital settings.

## Designing for Equity and Accessibility

Designing for equity and accessibility ensures that technology can be used by everyone, including people with different abilities and backgrounds.

Equal access means that every person should have the same opportunity to use technology and the internet. This includes access to reliable devices, internet connections, and digital skills. Some people may face challenges such as disabilities, language barriers, or lack of resources. Understanding these issues helps developers and organizations design solutions that reach everyone. Equal access ensures that no one is left behind in the digital world. Accessible design; benefits people with disabilities improves usability for everyone.



**CLASS ACTIVITY**

Groups complete a small task, such as planning a poster, while practicing clear online-style messaging.

### Creating Inclusive Applications for Diverse Users

Inclusive applications are designed to meet the needs of different types of users. This can include adding features such as screen readers for the visually impaired, subtitles for videos, or multiple language options. Simple layouts and clear text also make apps easier to use for beginners. Developers should test their designs with a variety of users to ensure accessibility for all. Creating inclusive digital tools supports equality and improves the overall user experience.

## Strategies for Effective Collaboration

Effective collaboration helps teams work smoothly and complete tasks successfully. Teamwork becomes stronger when everyone understands their responsibilities and communicates clearly. Planning, organizing, and sharing ideas allow group projects to move forward without confusion. Collaboration also teaches us how to respect different opinions and work toward a common goal. The following are common strategies for effective collaboration.



**TIDBIT**

Clear roles, task management, and goal setting improve teamwork efficiency.

**Assigning Roles**

Clear roles help each team member know exactly what they need to do. One person might handle research, another might organize information, while someone else prepares the final presentation.

**Managing Tasks**

Managing tasks by creating checklists or timelines helps the team stay on track and avoid delays.

**Setting Team Goals**

Setting shared goals gives everyone a direction to work toward and encourages cooperation. When roles and goals are clear, teamwork becomes much easier and more efficient.

**Reflecting on Collaborative Experiences**

After completing a project, it is helpful for students to think about what went well and what could be improved. Reflecting allows team members to identify strengths, such as good communication, and weaknesses, such as missed deadlines.

**Discussion to Improve Teamwork**

Discussing these experiences helps students learn from mistakes and become better team players. Reflection also encourages respect and understanding among group members. Over time, this practice leads to stronger teamwork and more successful collaborations.

## 9.4 Digital Entrepreneurship and Innovation

Digital entrepreneurship is the use of technology and online platforms to create, manage, and grow business ideas. It helps individuals solve real-world problems through innovative digital solutions. Entrepreneurs use social media, e-commerce websites, and freelancing platforms to reach customers globally. Innovation plays a key role in developing new products and services that meet changing market needs. This field provides opportunities for students to turn creative ideas into successful digital businesses.

**DO YOU KNOW?**

Most successful entrepreneurs are not born risk-takers—they develop risk-taking skills through experience.

### Characteristics of Successful Entrepreneurs

Successful entrepreneurs are creative and always look for new ideas to solve problems. They are hardworking and remain committed to achieving their goals. Good entrepreneurs are confident and willing to take calculated risks. They have strong decision-making and problem-solving skills. They are also adaptable and can adjust to changing market conditions and customer needs.

### Entrepreneurship in Pakistan and the Digital Economy

Entrepreneurship in Pakistan is growing rapidly due to increased access to the internet and mobile technology. Many young people are starting online businesses through social media, e-commerce platforms, and freelancing websites. The digital economy is creating new job opportunities in areas such as software development, digital marketing, and online services. Government and private initiatives are also supporting startups and innovation. This growth is helping improve the country's economy and encouraging youth to become job creators instead of job seekers.

### Identifying Business Opportunities

Identifying business opportunities means finding problems in society that can be solved through products or services. Entrepreneurs observe daily life to understand customer needs and market demands. Technology helps in discovering new opportunities such as online education, e-commerce, and digital services. Successful entrepreneurs focus on solving real-world problems in innovative ways. In Pakistan, areas like agriculture, healthcare, and education provide many opportunities for digital business ideas.



Sig9 SideB Process Black 07/17/2026 Computer 12 pecta ishaq

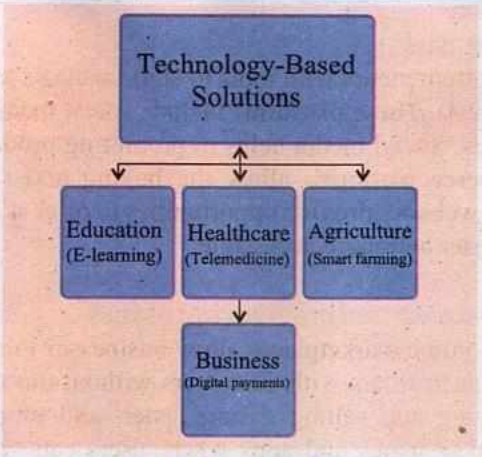



**Figure 9.4: Technology-Based Solutions**

## Solving Real-World Problems Through Technology

Technology helps entrepreneurs solve real-world problems in faster and more efficient ways. It allows people to create digital solutions for challenges in education, healthcare, agriculture, and business (as shown in Figure 9.4). For example, online learning platforms help students study from home, while mobile apps support farmers in improving crop production. Digital payment systems make financial transactions easier and safer. By using technology creatively, entrepreneurs can develop innovative solutions that improve people's daily lives.

## Recognizing Market Needs and Customer Demands

Recognizing market needs and customer demands means understanding what people want and what problems they face. Entrepreneurs study customer behavior, preferences, and trends to design better products and services. Surveys, feedback, and online data help identify gaps in the market. Meeting customer needs effectively leads to customer satisfaction and business success. In the digital economy, analyzing online trends and user feedback is very important for creating successful business ideas.

## Digital Platforms for Entrepreneurship

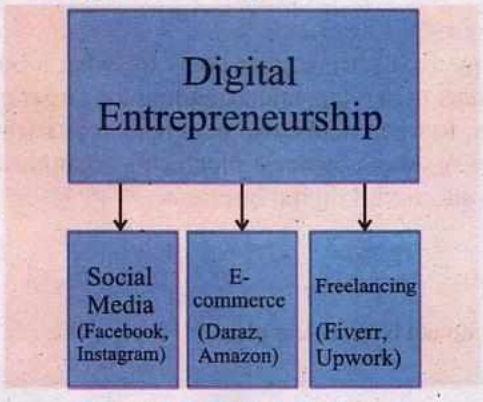



**Figure 9.5: Digital Platforms**

## Using Social Media for Business Promotion

Digital platforms provide entrepreneurs with tools to start, manage, and grow their businesses online (as shown in Figure 9.4). These platforms include social media, e-commerce websites, and freelancing marketplaces. Social media helps in promoting products and reaching a large audience quickly. E-commerce platforms allow the buying and selling of goods without physical shops. Freelancing websites provide opportunities to offer skills and services globally, making entrepreneurship easier and more accessible.

## E-commerce Platforms and Online Marketplaces

E-commerce platforms and online marketplaces allow businesses to sell products and services over the internet. They connect sellers with customers without the need for a physical store. These platforms make buying and selling easier, faster, and more accessible to a wider audience. Examples include websites and apps where users can browse products, compare prices, and place orders online. They play an important role in supporting digital entrepreneurship by reducing costs and expanding market reach.

## Freelancing and Remote Work Opportunities

Freelancing and remote work allow individuals to offer their skills and services to clients through online platforms. Freelancers can work in areas such as graphic design, writing, programming, and digital marketing. Remote work enables people to work for companies from home or any location with an internet connection. These opportunities help students and professionals earn income independently. They are an important part of digital entrepreneurship as they provide flexibility and global access to job markets.

## Planning a Small Digital Business

Planning a small digital business involves organizing ideas into a clear and workable plan. Entrepreneurs first identify a product or service that meets customer needs. They then decide on the target audience, pricing, and method of delivery using digital platforms. A basic budget is prepared to manage costs and expected income. Proper planning helps ensure the business is practical, organized, and more likely to succeed in the digital market.

## Basic Business Planning Concepts

Basic business planning involves setting clear goals for what a business wants to achieve. It includes identifying a product or service, understanding the target market, and deciding how to deliver value to customers. Entrepreneurs also plan resources such as time, money, and skills needed to run the business. A simple business plan helps organize ideas and provides direction for starting and growing a successful digital business.

**DO YOU KNOW?**

Most failed startups do not have a clear plan.



Sig9 SideA Process Black 07/17/2026 Computer 12. pedactishaq

## Budgeting and Resource Management

Budgeting and resource management involve planning how money, time, and other resources will be used in a business. A budget helps entrepreneurs estimate costs, set prices, and manage income and expenses effectively. Proper resource management ensures that materials, skills, and tools are used efficiently without waste. In digital businesses, good budgeting and resource planning help maintain profitability and support long-term success.

## Teamwork and Communication in Startups

Teamwork and communication are essential for the success of startups and small digital businesses. Team members work together by sharing ideas, responsibilities, and skills to achieve common goals. Effective communication helps avoid misunderstandings and improves decision-making. In startups, clear coordination through meetings, messaging apps, and online tools ensures smooth workflow. Strong teamwork and communication increase productivity and help the business grow successfully.

## Innovation and Future Opportunities

Innovation means creating new and improved ideas, products, or services that solve problems in better ways. In digital entrepreneurship, innovation helps businesses grow and stay competitive in the market. Future opportunities are expanding rapidly due to technologies such as artificial intelligence, mobile apps, and online platforms. These technologies are creating new career paths and business ideas for young entrepreneurs. By adopting innovation, individuals can build successful and future-ready digital businesses.

## Emerging Technologies and Entrepreneurship

Emerging technologies such as artificial intelligence, cloud computing, blockchain, and mobile applications are transforming entrepreneurship. These technologies help entrepreneurs create smarter, faster, and more efficient business solutions. They reduce costs, improve productivity, and enable global reach through digital platforms. Startups use these technologies to develop innovative products and services that meet modern customer needs. As a result, emerging technologies are opening new opportunities for digital entrepreneurship worldwide.

## Role of AI, Apps, and Digital Services in Business

Artificial Intelligence (AI), mobile apps, and digital services play an important role in modern business growth. AI helps entrepreneurs analyze data, understand customer behavior, and improve decision-making. Mobile apps make it easier for businesses to reach customers, sell products, and provide services quickly. Digital services such as online payments, cloud storage, and marketing tools improve efficiency and reduce costs. Together, these technologies support innovation and make businesses more competitive in the digital economy.

**DO YOU KNOW?**

AI tools can now design, write, and even help run businesses.

## Career Pathways in Technology Entrepreneurship

Technology entrepreneurship offers a variety of career pathways for students in the digital age. Individuals can become startup founders by developing and launching their own digital businesses. Others may work as freelancers in fields such as programming, graphic design, and digital marketing. Careers in app development, e-commerce management, and online consultancy are also growing rapidly. With the rise of technology, students can also become innovators who create new digital solutions for real-world problems.

# 9.5 Ethical, Legal, and Environmental Impacts of Computing

Computing affects many aspects of daily life. Technology can have ethical or unethical effects, legal or illegal consequences, and can impact the environment. Ethical use involves making fair and responsible choices when using digital tools. Legal use follows the laws related to technology. Environmental impact includes energy consumption, electronic waste, and other effects on nature. Awareness of these impacts helps make informed and responsible decisions when using computing resources.

## Ethical Use of Technology

Ethical use of technology involves acting responsibly and avoiding harm through digital actions. This includes respecting privacy, fairness, and proper handling of information. Following ethical guidelines helps maintain trust and safety online. Decisions in technology should consider the potential effects on people, organizations, and society. Responsible use of digital tools supports a safe and fair digital environment.

**DO YOU KNOW?**

Copying online content without permission can violate both copyright laws and digital ethics

### Respecting Intellectual Property and Data Ownership

Intellectual property refers to creative work such as software, music, or writing. Data ownership refers to the right to control personal information. Respecting these rights means avoiding copying, sharing, or using work without permission. Personal and sensitive information should be protected and not misused. Legal and ethical respect for intellectual property and data encourages fairness and protects creators and users.

### Understanding Consequences of Unethical Digital Behavior

Unethical digital behavior can include hacking, spreading false information, or cyberbullying. Such actions can cause harm, damage reputations, and lead to legal consequences. Even small actions online may have significant effects. Understanding the outcomes of unethical behavior encourages responsible digital conduct. Maintaining ethical standards ensures a safer and more trustworthy online environment.

## Legal Aspects and Data Privacy Laws

Legal aspects of computing refer to the rules and regulations that govern the use of technology and digital information. Cybercrime refers to illegal activities carried out using computers, networks, or the internet, such as hacking, identity theft, and online fraud. These laws protect individuals, organizations, and society from misuse of data and technology. Data privacy laws

ensure that personal and sensitive information is collected, stored, and shared responsibly. Understanding these laws helps prevent legal issues and promotes safe digital practices. Compliance with legal frameworks also encourages trust in online systems and services.

**Overview of International Frameworks (e.g., GDPR)**

The General Data Protection Regulation (GDPR) is a law in the European Union that protects personal data. It sets rules on how organizations collect, store, and use data. GDPR requires transparency, consent, and the right to access or delete personal information. Many countries and companies follow similar standards to ensure data privacy globally. These frameworks aim to prevent misuse of personal information and promote accountability.

**Cybercrime Laws and Data Protection Policies in Pakistan**

Pakistan has laws to prevent cybercrime and protect digital information. The Prevention of Electronic Crimes Act (PECA) 2016 is the main law in Pakistan that deals with cybercrime. It aims to protect individuals, organizations, and the state from illegal activities carried out using computers, mobile devices, and the internet. These laws cover hacking, identity theft, online fraud, and unauthorized access to data. Organizations are required to follow data protection policies to secure sensitive information. Legal penalties exist for violating cybercrime and data privacy laws. Following these regulations helps maintain security and trust in digital systems within the country.

**CLASS ACTIVITY**

Students list cybercrimes covered by PECA through a guided worksheet.

## Environmental and Social Impacts

Computing affects the environment and society in many ways. Technology can consume energy, produce electronic waste, and contribute to pollution. At the same time, technology influences social behavior, communication, and culture. Understanding environmental and social impacts helps promote responsible use of digital tools. Sustainable practices and awareness of societal effects are important for balancing technology with nature and communities.

**E-waste Management and Sustainable Computing**

Electronic waste (e-waste) is created when devices are discarded or replaced. Improper disposal of e-waste can harm the environment and human health. Sustainable computing includes recycling, reusing, and reducing electronic devices responsibly. Energy-efficient devices and environmentally friendly practices reduce the negative impact of technology. Proper e-waste management ensures resources are preserved and pollution is minimized.

**TIDBIT**

Recycling, reusing, and energy-efficient devices reduce environmental harm.

## The Role of Technology in Cultural and Societal Change

Technology changes how people communicate, work, and access information. Social media, digital learning, and online services influence culture and daily life. Technology can improve

education, healthcare, and business efficiency. However, it can also create challenges such as digital divide or loss of privacy. Understanding these social impacts encourages thoughtful and responsible use of technology in communities.

## Summary

**Cybersecurity**
Cybersecurity is the practice of protecting computers, networks, data, and online accounts from unauthorized access, misuse, damage, or cyberattacks.

**Online Safety**
Online safety means using the Internet carefully and responsibly to protect personal information, devices, and digital identity.

**Strong Password**
A strong password is a secure password that is difficult to guess. It usually includes a combination of uppercase letters, lowercase letters, numbers, and symbols.

**Privacy Settings**
Privacy settings are options on digital platforms that allow users to control who can see their personal information, posts, profile, location, and online activity.

**Privacy Policy**
A privacy policy explains how an application, website, or organisation collects, uses, stores, and shares user data.

**Digital Collaboration**
Digital collaboration means working together online using tools such as Google Workspace, Microsoft Teams, shared documents, video meetings, and group chats.

**Digital Etiquette**
Digital etiquette refers to polite, respectful, and responsible behaviour while communicating online.

**Cyber Threat**
A cyber threat is any possible danger to digital systems, data, or users. Examples include phishing, malware, ransomware, hacking, and identity theft.

**Phishing**
Phishing is a cyberattack in which fake emails, messages, or websites trick users into sharing passwords, bank details, or personal information.

**Malware**
Malware is harmful software designed to damage a computer, steal data, or disturb normal system operations.

**Ransomware**
Ransomware is a type of malware that locks files or systems and demands payment to restore access.

**Two-Factor Authentication**
Two-Factor Authentication (2FA) is a security method that requires two forms of verification, such as a password and a code sent to a mobile phone.

**Biometric Verification**

Biometric verification uses unique physical features, such as fingerprints, face recognition, or voice recognition, to confirm a user's identity.

**Firewall**

A firewall is a security tool that monitors and controls incoming and outgoing network traffic according to security rules.

**Antivirus Software**

Antivirus software is a program used to detect, block, and remove viruses and other harmful software from a computer.

**Cryptography**

Cryptography is the technique of protecting information by converting it into a secure form so that only authorized users can read it.

**Data Protection**

Data protection means keeping personal, academic, organizational, or sensitive information safe from loss, misuse, theft, or unauthorized access.

**Accessibility**

Accessibility means designing digital systems so that people with different abilities, needs, and backgrounds can use them easily.

**Intellectual Property**

Intellectual property refers to creations of the mind, such as software, books, images, videos, designs, and inventions, that are legally owned by individuals or organisations.

**Data Privacy Laws**

Data privacy laws are legal rules that protect how personal data is collected, stored, used, and shared.

**E- Waste**

E- waste means discarded electronidevices, such as old computers, mobile phones, batteries, printers, and chargers.

**Sustainable Computing**

Sustainable computing means using, designing, and disposing of technology in a way that reduces harm to the environment.

**Entrepreneurship**

Entrepreneurship is the process of creating innovative business ideas, solving problems, and developing opportunities using technology and digital platforms.

**Digital Entrepreneurship**

Digital entrepreneurship involves using the Internet, social media, e-commerce, or digital tools to create and manage businesses or services.

# EXERCISE

## Multiple Choice Questions

1. The following helps keep online accounts secure:
   (a) Weak passwords (b) Strong passwords
   (c) Sharing passwords (d) Simple passwords
2. The main danger of downloading unknown software is:
   (a) Faster browsing (b) Getting rewards
   (c) Malware infection (d) Extra storage
3. The tool that adds an extra layer of login security is:
   (a) Screensaver (b) Two-Factor Authentication
   (c) Wallpaper (d) Search engine
4. The cyber threat that tricks users into giving personal information is:
   (a) Phishing (b) Antivirus
   (c) Encryption (d) Backup
5. The purpose of a firewall is to:
   (a) Delete files (b) Block harmful network traffic
   (c) Play games (d) Slow down the computer
6. Encryption is used to:
   (a) Delete data (b) Make data readable to everyone
   (c) Convert data into a secure form (d) Upload data automatically
7. A commonly used platform for online collaboration is:
   (a) Paint (b) Google Workspace
   (c) Notepad (d) VLC Player
8. Digital etiquette means:
   (a) Sharing all files freely (b) Using rude language online
   (c) Safe and polite online behaviour (d) Ignoring team members
9. E-waste means:
   (a) Old clothes (b) Electronic garbage
   (c) Raw materials (d) Food waste
10. The international law that protects user privacy is:
    (a) HTML (b) GDPR
    (c) CPU (d) LAN

## Short Questions

1. What is the role of strong passwords in protecting data?
2. How can downloading suspicious software affect a computer?
3. Why is it important to install software updates and security patches?
4. What are common types of cyber threats, such as phishing or ransomware?

5. How does Two-Factor Authentication (2FA) improve online security?
6. Who is responsible for managing privacy and security settings on online platforms?
7. What is intellectual property, and why should it be respected in digital use?
8. How does e-waste impact the environment, and what is sustainable computing?
9. Where do international data protection laws like GDPR apply?
10. What is entrepreneurship?

## Long Questions

1. Explain the importance of online safety and describe how strong passwords and safe browsing habits protect personal data.
2. Discuss the role of software updates and security patches in maintaining computer security, and explain the risks associated with outdated software.
3. Describe how privacy and security settings on online platforms can be managed. Explain why reading privacy policies is important for data protection.
4. Explain different types of cyber threats such as viruses, phishing, ransomware, spyware, and DDoS attacks. Discuss how these threats can affect both individuals and organizations.
5. Discuss security tools and techniques like Two-Factor Authentication (2FA), biometric verification, firewalls, intrusion detection systems (IDS), and antivirus software. Explain how each helps in protecting data.
6. Explain the concept of entrepreneurship and describe how digital technologies have created new opportunities for entrepreneurs. Include at least three examples of digital entrepreneurship.
7. Explain the legal aspects of computing by discussing international frameworks like GDPR and cybercrime laws and data protection policies in Pakistan. How do these laws protect digital information?
8. Discuss the environmental and social impacts of computing. Explain how e-waste management and sustainable computing practices can reduce harm to the environment, and describe the role of technology in cultural and societal change.

## Answer Key

1. B
2. C
3. B
4. A
5. B
6. C
7. B
8. C
9. B
10. C

# Instructions for Preparation of Exam Paper of Computer Science and Entrepreneurship for Class XII

## ESSENTIAL INSTRUCTIONS FOR PAPER SETTERS

The paper of Computer Science and Entrepreneurship for class XII will consist of 75 marks. Timing of the paper will be 2:30 hours. The paper will be made as per following details:

| | | |
|---|---|---|
| **Part-I:**<br>**Objective:** | **Q-1:**<br>15 Multiple Choice Questions will be selected from the entire contents of textbooks; the distribution will be as under:<br>**Ch #** **MCQ**<br>01 ---------- 02<br>02 ---------- 02<br>03 ---------- 02<br>04 ---------- 02<br>05 ---------- 02<br>06 ---------- 02<br>07 ---------- 01<br>08 ---------- 01<br>09 ---------- 01 | $1 \times 15 = 15$ |
| **Part-II:**<br>**Subjective:** | This section will contain three questions i,e 2,3,4 and each question contain 9 short questions. The distribution will be as under:<br>**Q-2: Attempts (6 out of 9)**<br>**Ch #** **SQ**<br>01 ---------- 03<br>02 ---------- 04<br>06 ---------- 02 | $2 \times 6 = 12$ |
| | **Q-3: Attempts (6 out of 9)**<br>**Ch #** **SQ**<br>03 ---------- 02<br>04 ---------- 04<br>05 ---------- 03 | $2 \times 6 = 12$ |

| | | |
|---|---|---|
| | **Q-4: Attempts (6 out of 9)**<br>**Ch #** **SQ**<br>07 ..... 05<br>08 ..... 02<br>09 ..... 02 | 2 × 6 = 12 |
| **Part-III: Subjective:** | This section will contain five long questions and students have to attempt any 3 questions out of5 questions. Each question carries 8 marks. The detail is as under:<br>**Q-5:**<br>**Ch #** **Question**<br>01 ..... 01<br>**Q-6:**<br>**Ch #** **Question**<br>03 ..... 01<br>**Q-7:**<br>**Ch #** **Question**<br>06 ..... 01<br>**Q-8:**<br>**Ch #** **Question**<br>08 ..... 01<br>**Q-9:**<br>**Ch #** **Question**<br>09 ..... 01 | 3 × 8 = 24 |

# Model Paper Computer Science and Entrepreneurship
# Intermediate part-II

## OBJECTIVE

**Question 1: Multiple Choice Questions (MCQ)**

Encircle the correct answer. Four possible answers A,B,C,D are given to each question. The choice which you think is correct, fill that circle in front of that question with marker or pen ink in the answer book. Cuttings are filling two or more circles will result in zero mark in that question.

**Total Marks: 15** **Time Allowed: 20 mins**

1. Which protocol is mainly used to view web pages?
   a. FTP b. DNS c. TCP d. HTTP
2. Which device converts digital signals for internet transmission?
   a. Switch b. Hub c. Modem d. NIC
3. Which of the following is a compound proposition?
   a. P b. Q c. P AND Q d. True
4. A proposition that is true for at least one case is called:
   a. Unsatisfiable b. Invalid c. Satisfiable d. Equivalent
5. Which concept hides data inside a class?
   a. Inheritance b. Polymorphism c. Encapsulation d. Abstraction
6. Which keyword is used to define a class in Python?
   a. def b. class c. function d. object
7. Which of the following is Python's built-in GUI toolkit?
   a. wxPython b. Tkinter c. PyQt d. Kivy
8. What widget is used to get user input in Tkinter?
   a. Label b. Button c. Entry d. Listbox
9. Which ython module is used for unit testing?
   a. numpy b. matplotlib c. unitest d. os
10. Which Python keyword is used for exception handling?
    a. try-except b. try-finally c. try-catch d. handle-throw
11. Which metric measures overall correctness of a model?
    a. Precision b. Recall c. Accuracy d. F1-score
12. Machine learning allows systems to:
    a. Work without data b. Learn from data and improve performance
    c. Only store data d. Replace databases
13. Which of the following is a test statistic?
    a. Mean b. t-value c. Mode d. Median
14. Which of the following is an example of Cloud Computing?
    a. AWS b. MS-Word c. Smartwatch d. USB flash drive
15. What is the main danger of downloading unknown software?
    a. Faster browsing b. Malware infection
    c. Getting rewards d. Extra storage

# Model Paper Computer Science and Entrepreneurship
## Intermediate part-II
## SUBJECTIVE

**Total Marks: 60** **Time Allowed: 2:10 hours**

### SECTION I

**Q # 2: Write short answers of any six questions from the following: (6*2=12)**

i. What is network architecture?
ii. A network architecture experiences a single cable break, and the entire network goes down. Which topology is being used, and why did this failure happen?
iii. How does **DNS** (Domain Name System) function like a phonebook for the internet?
iv. Why is propositional logic insufficient for a statement like "All students love coding," making predicate logic necessary instead?
v. What it means when a compound statement is called a tautology?
vi. Formulate a predicate logic statement using variables, a predicate, and a quantifier to represent the sentence: "Someone in this classroom owns a laptop."
vii. Break down the task of "cleaning a messy bedroom" into three smaller, manageable sub-tasks (Decomposition).
viii. What is the primary difference between Supervised Learning and Unsupervised Learning?
ix. Why do we split our data into a training set and a testing set instead of using all of it for training?

**Q # 3: Write short answers of any six questions from the following: (6*2=12)**

i. How does encapsulation help in keeping data secure and private?
ii. If you have a class called **Player** for a video game, write a single line of code to add an attribute for the player's health and set it to 100.
iii. You want to clear a text box after a user clicks "Submit". Which **Tkinter** method helps you delete the text inside an **Entry** widget?
iv. What is the purpose of the **.mainloop()** function in a **Tkinter** program?
v. Write a single line of **Tkinter** code to create a window named **root**.
vi. When a user clicks a "Save" button, your program crashes with an error saying the database table does not exist. Analyze the problem: what step did the programmer forget to do first?
vii. In debugging, what is a **breakpoint**?
viii. Explain the difference between a **Syntax Error** (like a typo in code) and an **Exception.**
ix. Look at this code: **result = 10 / user_input.** If the user types 0, what specific type of built-in Python exception will be raised?

**Q # 4: Write short answers of any six questions from the following: (6*2=12)**

i. What is the main difference between the Null Hypothesis ($H_0$) and the Alternative Hypothesis ($H_a$)?

ii. Why is it considered an ethical issue to completely delete "outliers" (unusual data points) from your dataset just to make your hypothesis look correct?

iii. If you reject the Null Hypothesis at a 5% significance level ($\alpha = 0.05$), what is the percentage chance that you might be making a mistake and rejecting a true hypothesis?

iv. Draft a clear Null Hypothesis ($H_0$) and Alternative Hypothesis ($H_a$) for an experiment testing whether rolling a standard 6-sided die is fair or biased.

v. Why is the p-value important in hypothesis testing?

vi. How Artificial Intelligence is useful in our daily lives? Give two examples.

vii. How does blockchain ensure security and transparency in transactions?

viii. How does e-waste impact the environment?

ix. How can downloading suspicious software affect a computer?

## SECTION II

**Note: Attempt any three descriptive answers of the following questions: (3*8=24)**

Q. 5: What is OSI model? Explain its layers and their functions? **(08)**

Q. 6: Write a Python program to demonstrate multilevel inheritance using Grandfather, Father, and Son. **(08)**

Q. 7: Describe model evaluation techniques and explain accuracy, precision, recall and F1-score. **(08)**

Q. 8: Discuss the benefits and challenges of Artificial Intelligence (AI) in society. How can ethical principles ensure fair and responsible use of AI? **(08)**

Q. 9: Describe how privacy and security settings on online platforms can be managed. Explain why reading privacy policies is important for data protection? **(08)**

ii. Why is it considered an ethical issue to completely delete "outliers" (unusual data points) from your dataset just to make your hypothesis look correct?

iii. If you reject the Null Hypothesis at a 5% significance level ($\alpha = 0.05$), what is the percentage chance that you might be making a mistake and rejecting a true hypothesis?

iv. Draft a clear Null Hypothesis ($H_0$) and Alternative Hypothesis ($H_a$) for an experiment testing whether rolling a standard 6-sided die is fair or biased.

v. Why is the p-value important in hypothesis testing?

vi. How Artificial Intelligence is useful in our daily lives? Give two examples.

vii. How does blockchain ensure security and transparency in transactions?

viii. How does e-waste impact the environment?

ix. How can downloading suspicious software affect a computer?

## SECTION II

**Note: Attempt any three descriptive answers of the following questions: (3*8=24)**

Q. 5: What is OSI model? Explain its layers and their functions? **(08)**

Q. 6: Write a Python program to demonstrate multilevel inheritance using Grandfather, Father, and Son. **(08)**

Q. 7: Describe model evaluation techniques and explain accuracy, precision, recall and F1-score. **(08)**

Q. 8: Discuss the benefits and challenges of Artificial Intelligence (AI) in society. How can ethical principles ensure fair and responsible use of AI? **(08)**

Q. 9: Describe how privacy and security settings on online platforms can be managed. Explain why reading privacy policies is important for data protection? **(08)**



# قومی ترانہ

پاک سَرزمین شاد باد　　کِشورِ حسین شاد باد
تُو نِشانِ عزمِ عالی شان　　ارضِ　　پاکستان
مرکزِ یقین شاد باد

پاک سَرزمین کا نِظام　　قُوّتِ اُخوّتِ عوام
قوم ، مُلک ، سلطنت　　پایندہ تابندہ باد
شاد باد منزلِ مُراد

پَرچمِ ستارہ و ہِلال　　رہبرِ ترقّی و کمال
ترجمانِ ماضی، شانِ حال　　جانِ　　اِستقبال
سایۂ خدائے ذُوالجلال

PECTAA
Transformation, Innovation and Excellence

PUNJAB EDUCATION, CURRICULUM, TRAINING
AND ASSESSMENT AUTHORITY