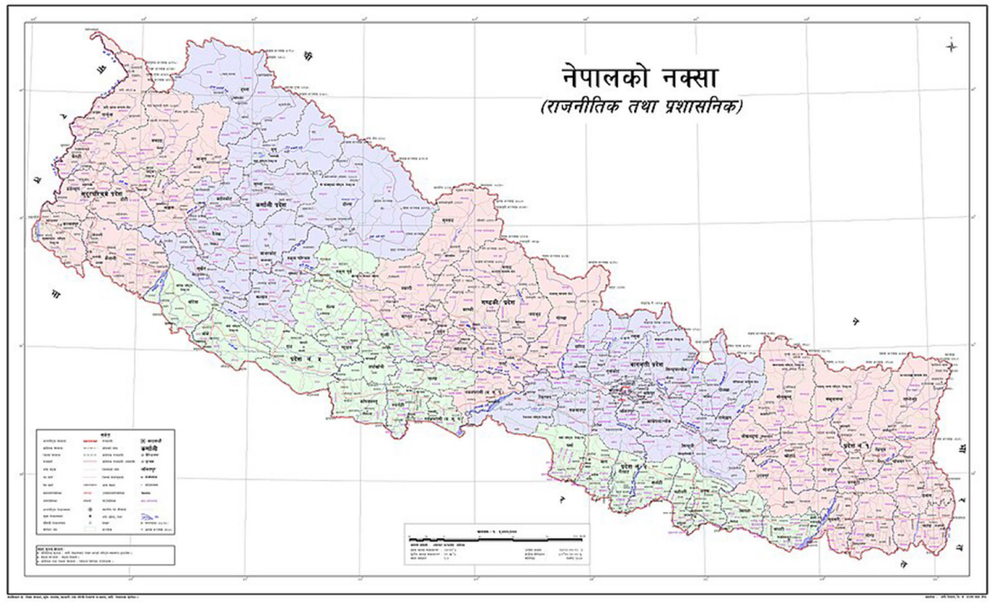


नेपाल सरकार
शिक्षा, विज्ञान तथा प्रविधि मन्त्रालय
पाठ्यक्रम विकास केन्द्र
सानोठिमी, भक्तपुर

Computer Science, Class 10

Curriculum Development Center

COMPUTER SCIENCE

Class 10

COMPUTER SCIENCE

Class 10

Government of Nepal
Ministry of Education, Science and Technology
Curriculum Development Centre
Sanothimi, Bhaktapur

Publisher :

Government of Nepal
Ministry of Education, Science and Technology
Curriculum Development Centre
Sanothimi, Bhaktapur

ISBN:

First Edition : 2082 B.S.

Printed at :

Price :

Preface

A textbook is a main tool for the use of students on a particular subject that is usually developed based on a curriculum to meet its curriculum standards. This 'Computer Science, Grade 10' textbook has been developed to address the main aims of Secondary Education. It aligns with the intent pursued by the National Curriculum Framework of School Education, 2076 and has been fully developed in accordance with the Secondary Level Optional Computer Science Curriculum, 2081. All components of each lesson in the textbook are equally important and they have the specific aim of presentation so as to fulfill the learning outcomes stated in the curriculum.

This textbook is prepared by a team that includes Mr. Sunil Kumar Gharti, Mr. Bimal Thapa, Mr. Shyam Kumar Adhikari, Mr. Rajendra Rokaya, Mr. Binay Adhikari, Mr. Pratik Man Singh and Mr. Lav Dev Bhatta. Several people notably, the Director General, Mr. Yubaraj Paudel, the Subject Committee Chairperson, Dr. Bed Prasad Dhakal and the Subject Committee members; Dr. Bal Krishna Subedi, Mrs. Maiya Khadka and Mr. Purushottam Ghimire contributed a lot in framing the book in this form. The contribution made by other experts and the practitioners; Mr. Rom Kanta Pandey, Mrs. Salina Dangol is appreciated. The language of the book was edited by Mr. Nabin Kumar Khadka and Deepak Raj Bhatta. The layout of the book was designed by Mr. Bhakta Bahadur Karki. The Curriculum Development Centre extends sincere gratitude to all of them.

This textbook may be used to deliver the foundation for the content of the curriculum. However, the balance between theoretical and practical aspects is to be maintained by the teachers. This textbook can be used as the primary resource for classroom teaching but teachers are encouraged to make use of their own resources to supplement the computer science learning of their students. Curriculum Development Centre always welcomes constructive feedback for the betterment of its publications.

2025 AD

Curriculum Development Centre
Sanothimi, Bhaktapur

Contents

Unit 1 Computer Network and Communication

Before you read

Look at the pictures and discuss the following questions.

Figure 1.1 Network and Communication

a. *What is a computer network? Why is it important in today's digital world?*

b. *Why is it important to choose the right network topology for an organization?*

c. *How do 4G LTE and 5G technologies improve network performance and reliability?*

A **computer network** is a group of two or more computers and devices that are connected to share data, resources, and messages. These connections can be made using wires (like Ethernet cables) or without wires (like Wi-Fi and Bluetooth). Today, networks are very important because they help us send emails, make video

calls, share files, use the Internet, study online, and work from home. Networks also help smart devices like home assistants and sensors work properly.

Mobile networks like 4G **LTE** (Long Term Evolution) and 5G make the Internet faster and more reliable. 4G LTE gives better speed for video calls, online classes, and downloading. 5G is even faster, with almost no delay and can connect many devices at once without slowing down. It works well even in busy places and gives a strong, stable connection. This helps everyone use mobile Internet easily and quickly.

1.1 Concept of Telecommunication

Telecommunication means sending and receiving information over long distances with the help of electronic devices. These devices include telephones, mobile phones, radios, televisions, and computers. For example, when you make a phone call, send a message, or watch a TV program, you are using telecommunication.

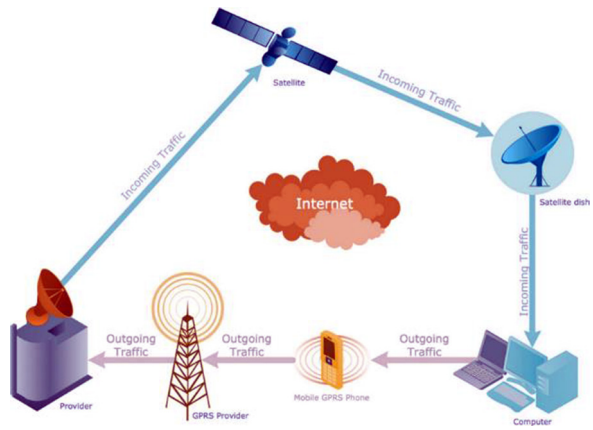


Fig 1.2 Telecommunication Devices

To make this communication possible, different devices are connected together. This connection is called a **network**. A network can use wires (like telephone lines or cables) or work wirelessly (like Wi-Fi or mobile networks). Through these networks, we can share messages, sounds, pictures, and videos from one place to another quickly and easily.

Broadband

Broadband is a high-speed Internet connection that allows you to send and receive a large amount of data quickly. It is much faster than the old dial-up Internet used in the past. With broadband, you can easily watch videos, join online classes, play games, or download files without waiting too long.

There are different types of broadband connections, such as:

i. **DSL (Digital Subscriber Line)** – uses telephone lines.

ii. **Cable** – uses TV cables.

iii. **Fiber-optic** – uses special glass wires for very high speed.

iv. **Satellite** – connects through signals from satellites in space.

v. **Wireless** – uses mobile networks like 4G or Wi-Fi.

Broadband is commonly used in homes, schools, offices, and public places.

Throughput

Throughput means the actual amount of data that is successfully sent or received over a network in a certain amount of time. It shows how fast data is really moving between devices.

Throughput is usually measured in:

- **bps** – Bits per second
- **Mbps** – Megabits per second
- **Gbps** – Gigabits per second

Higher throughput means faster and smoother performance, like quicker downloads, video streaming without buffering, and better online experiences.

Bandwidth

Bandwidth is the maximum amount of data that a network can carry in a given time. It tells us how much data can be sent or received at once, like the size of a highway for data. It is measured in **bps, kbps** (kilobits per second), **Mbps** (Megabits per second) and **Gbps** (Gigabits per second).

Higher bandwidth means more data can travel at once, which gives faster internet speed helping you stream videos, download files, or join online classes more smoothly.

3G (Third Generation Network)

3G is the third generation of mobile network technology. It was a big improvement over older 2G networks. With 3G, people could use the Internet on their phones more easily and at faster speeds.

3G allows:

i. Faster Internet browsing
ii. Sending and receiving pictures and videos (multimedia messaging)
iii. Making video calls
iv. Using mobile broadband services like online games and apps

It helped bring the world closer by making smartphones more useful for learning, communication, and entertainment.

4G (Fourth Generation Network)

4G is the fourth generation of mobile network technology. It is much faster and more reliable than 3G. With 4G, using the Internet on mobile devices became smoother and quicker.

4G offers:

i. Very fast download and upload speeds
ii. Low latency (less delay during online activities)
iii. High-quality video streaming (HD videos)
iv. Better online gaming experience
v. Quick access to websites, apps, and cloud services

4G made it easier to study, play, and connect with others using mobile phones and tablets.

5G (Fifth Generation Network)

5G is the fifth generation of mobile network technology. It is much faster, smarter, and more powerful than 4G. It gives very high speed, strong connection, and can support many devices at the same time.

5G helps in:

i. Super-fast Internet access

ii. Running advanced apps like **Augmented Reality (AR)** and **Virtual Reality (VR)**

iii. Supporting **autonomous vehicles** (self-driving cars)

iv. Connecting smart devices in the **Internet of Things (IoT)**, like smart homes and cities

5G is changing the way we live, learn, travel, and use technology in many fields.

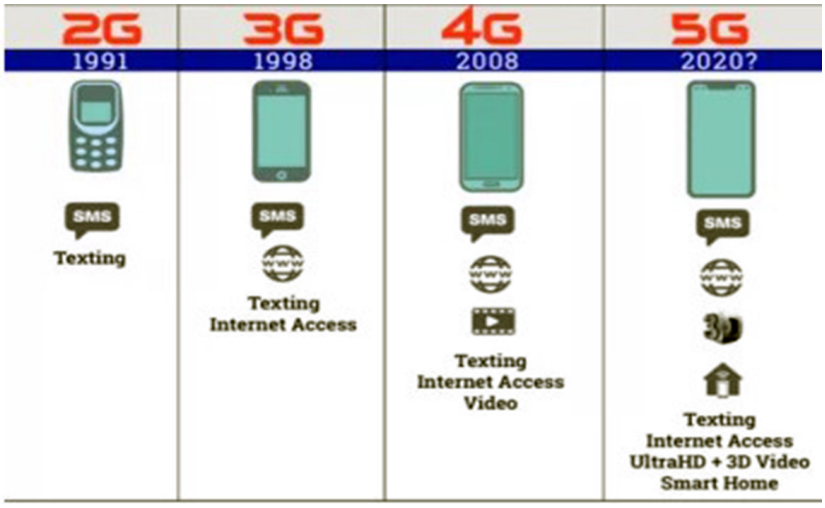


Figure 1.3 : 3G/4G/5G Network (Cell Phone)

Data Packets

When we send information over a computer network, it is broken into small parts called **data packets**. These packets travel across the network and are put back together at the destination.

Data packets help to:

- Send information faster
- Make data transfer more reliable
- Ensure efficient communication over long distances

Each packet carries a part of the message, along with details like the sender's and receiver's addresses, so it reaches the correct place.

Frequency

In telecommunication, **frequency** means how often data signals are sent over a communication channel in one second. It helps decide how fast and clearly signals can travel through the air or wires. Different technologies use different frequency ranges to send data.

Frequency is important for

i. Wi-Fi and Bluetooth connections

ii. Mobile networks like 4G and 5G

iii. Radio and TV broadcasting

iv. Devices like microwave ovens and RFID systems

Frequency is measured in

- **kHz** – kilohertz (1,000 times per second)
- **MHz** – megahertz (1 million times per second)
- **GHz** – gigahertz (1 billion times per second)

Examples of Frequency Use in Daily Life

Technology / Use	Frequency Range	Purpose / Use
Wi-Fi (2.4 GHz)	2.4 – 2.5 GHz	Wireless internet at home or school (WLAN)
FM Radio Broadcasting	88 – 108 MHz	Playing music and news on FM radio
Microwave Ovens	2.45 GHz	Heating food

Technology / Use	Frequency Range	Purpose / Use
Bluetooth	2.4 – 2.5 GHz	Connecting wireless devices (like speakers, earbuds)
AM Radio Broadcasting	530 – 1710 kHz	Long-distance radio, especially at night
5G Cellular (mmWave)	24 – 100 GHz	Very fast mobile communication in short distances
RFID (UHF)	860 – 960 MHz	Used in smart cards, inventory tracking
Television Broadcasting (UHF)	470 – 890 MHz	Digital TV signals

Did you know?

*The word **telecommunication** comes from the Greek "tele" (meaning **far off**) and Latin "communicare" (to **share**). So it means "**sharing from far away**."*

Communication Mode

Data communication is transferring data and information between computers and other electronic devices. The two types of data communication are **simplex mode and duplex mode.** Simplex mode is the transmission of data and information that takes place in only one direction. It is the unidirectional mode of data transmission. Radio, newspaper, television broadcasting, books, etc. are examples of simplex mode.

Figure 1.4 Simplex mode

Duplex mode is the mode of data transmission in which data and information flow in both directions. It is the bi-directional mode of data transmission. The two types of duplex modes are: **half duplex mode and full duplex mode.** Half duplex is a mode of data and information flows in both directions but only one direction at a time. Walky-talky and wireless handsets are examples of half-duplex modes.

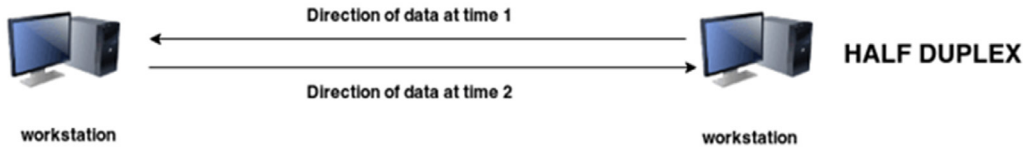


Figure 1.5 Half-Duplex Mode

In **full duplex mode**, data and information flow in both directions simultaneously on the transmission path. Mobile and landline phones are examples of full-duplex mode.

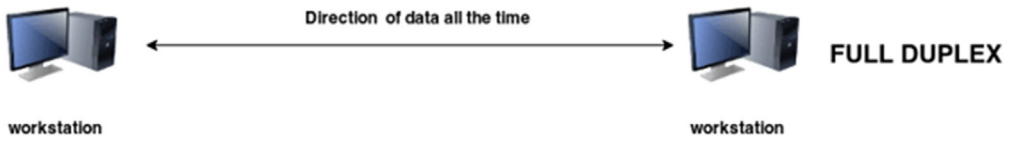


Figure 1.6 Full Duplex Mode

1.2 Communication Media

Communication media is the path or channel used to send data and information between devices in a network. It is also called **transmission media**. This media helps connect different devices so they can share messages and data.

There are two types of communication media:

i. **Guided Media** – uses physical wires or cables (like twisted pair, coaxial, and fiber optic).

ii. **Unguided Media** – uses wireless signals through air (like Wi-Fi, Bluetooth, radio waves).

These media can be either **wired** (like Ethernet cables) or **wireless** (like mobile networks), depending on how the devices are connected.

Guided Media

Guided communication media use **wires or cables** to transfer data and information between computers and other devices. This method is often **faster, safer, and more reliable** than wireless communication.

There are different types of cables used in networks, such as:

i. **CAT6 (Category 6)** – a common cable used for fast Ethernet connections.

ii. **Optical fiber** – a cable made of glass or plastic that transfers data using light. It is very fast and can carry data over long distances.

Guided media is also called **wired media** because it uses physical cables to connect devices.

a. **CAT6 (Category 6 cable): CAT6** is a type of network cable used to connect computers, routers, switches, and other devices to the Internet. It is faster and more reliable than older cables like **CAT5e**.

Key features of CAT6:

i. Supports speeds up to **1 Gbps** over 100 meters

ii. Can reach **10 Gbps** over shorter distances

iii. Has better insulation and tightly twisted wires

iv. Reduces signal interference for a stable connection

Figure 1.7 : CAT 6

CAT6 cables are commonly used in homes, offices, and data centers where **fast and strong internet** is important.

b. **Optical Fiber Cable:** It is a high-speed communication cable made of thin strands of plastic or glass. It uses **light waves** to transmit data signals from one

end to the other, allowing data to travel at very high speeds with minimal loss. Compared to twisted pair and coaxial cables, optical fiber is more powerful, reliable, and efficient. Because of its speed and quality, it is widely used by **Internet Service Providers (ISPs)** for providing fast internet services. Common connectors used with optical fiber cables include **Straight Tip (ST)**, **Screw-Mounted Adaptors (SMA)**, and **Subscriber Connector (SC)**.

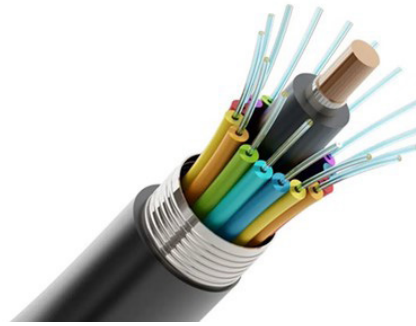


Figure 1.8 Optical fiber cable

Fun Facts

- *Speed of Light: Optical fiber sends data using light pulses, traveling at near light speed—over 200,000 km/s!*
- *Hair-Thin Power: These cables are thinner than a human hair but can carry terabytes of data over long distances!*
- *Bendable Glass: Made up of glass or plastic, they can be bent without breaking—amazing for something that's so fragile!*

Unguided Media

Unguided communication media, also known as **wireless communication**, transfers data and information **without using wires or cables**. Instead, it uses **radio waves or signals** to send and receive data between devices. Wireless technology is very important in today's communication systems, helping us connect to the Internet, share files, and use mobile services easily. In Nepal, permission to use wireless technology

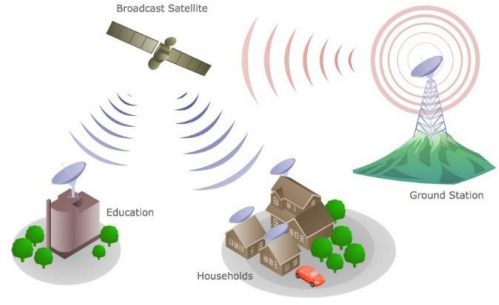


Figure 1.9 Unguided communication

is given by the government to ensure proper use and security.

Some common types of wireless communication are:

i. **Wi-Fi** – for wireless internet access

ii. **Bluetooth** – for short-range device connections

iii. **RFID** – for tracking and identification (like ID cards)

iv. **Satellite Communication** – for global communication using satellites

i. Wi-Fi (Wireless Fidelity): Wi-Fi is a wireless technology that allows devices to connect to the **Internet without using cables**. It uses **radio signals** to send and receive data and works on **2.4 GHz** and **5 GHz** frequency bands. Wi-Fi uses devices called **access points**, like **routers**, to connect phones, computers, tablets, and other gadgets to the Internet.

Figure 1.10 Wi-Fi Signal

To give a better and faster connection, Wi-Fi can select the **best channels** and use smart techniques like **channel bonding** (combining two channels) and **beamforming** (focusing signals toward a device). Wi-Fi is widely used in **homes, schools, offices, and public places** to provide Internet to many devices at the same time.

ii. Bluetooth: Bluetooth is a wireless technology that allows devices to share data over **short distances** without using any cables. It is commonly found in **smartphones, tablets, computers, smartwatches, and fitness bands**.

Bluetooth uses **radio waves** to connect devices and share files, music, or other information. It works using a smart method called **Frequency Hopping Spread Spectrum (FHSS)**, which helps avoid signal problems and makes the connection more reliable.

Bluetooth uses **very little power**, so it is good for small devices. Today, Bluetooth is built into **most modern electronic gadgets** and is useful for connecting headphones, keyboards, speakers, and more.

Figure 1.11 Bluetooth and Watch

iii. **Radio Frequency Identification (RFID)**: **Radio Frequency Identification** is a wireless technology that uses **radio waves** to transfer information between **RFID tags** and **readers**. RFID tags can be attached to objects, and when a reader scans them, it receives the stored data. These tags can be either **passive** (without a battery) or **active** (with a battery for longer range). RFID is widely used in many industries for **identifying and tracking items**, **managing inventory**, **controlling access** to secure areas, and even for **contactless payment systems**. It is a fast and reliable way to collect data and is used in stores, offices, schools, and transportation systems.

Figure 1.12 RFID

iv. **Satellite communication:** It is a method of sending information wirelessly across the world using **artificial satellites** placed in outer space. It has been in use since **1965** for services like **telecommunication, radio, television, and the Internet**. Using satellite communication, we can transfer **text, photos,**

videos, and more to any part of the world. Satellites work like **microwave relay stations in the sky**, receiving signals from one place and sending them to another. Today, many countries use satellites for different purposes, such as **TV broadcasting, weather forecasting, military communication, GPS navigation**, and **Internet services**.

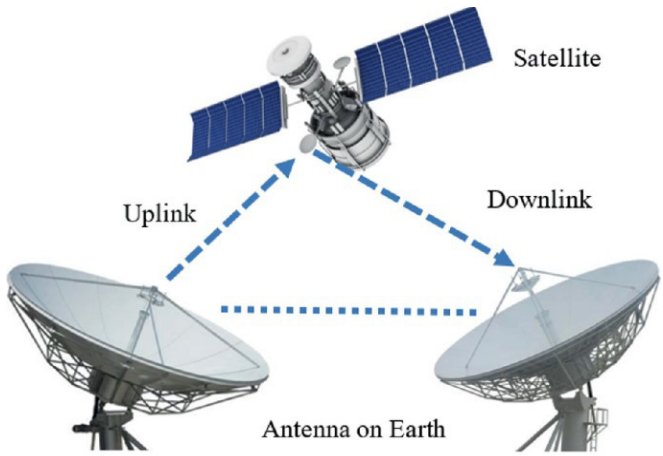


Figure 1.13 Satellite Communication

The comparison between guided and unguided communication media can be seen in the following table.

Guided media	Unguided media
Data and signals are sent through wires or cables.	Data and signals are sent through the air without wires.
Not affected by weather conditions like rain or wind.	Can be affected by weather conditions like rain or wind.
Used to connect close devices such as in a Local Area Network (LAN).	Used to connect devices far away such as in a Wide Area Network (WAN).
Provides a fixed path or direction for data.	No fixed path; data travels freely through the air.
Also called wired or guided media.	Also called wireless or unguided media.
Examples: CAT6 cable, Coaxial cable, Optical fiber.	Examples: Wi-Fi, Bluetooth, Microwave, Satellite, Infrared.

1.3 Connector

Connectors are small hardware devices used to connect **communication media** (like cables) to **network devices** such as computers, routers, and switches. They help transfer data signals between the devices and the network. Two common types of connectors are the **RJ-45 connector**, which is used for Ethernet cables, and the **media converter connector**, which is used to connect different types of cables, such as fiber optic and Ethernet, in a network system.

Registered Jack 45 (RJ-45) : The RJ45 connector is a standardized interface for connecting Ethernet cables to network devices like computers, routers, and switches. It features eight pins in a modular jack format, easy insertion and removal, and follows specific wiring schemes. It supports reliable, high-speed data transmission over Ethernet networks.

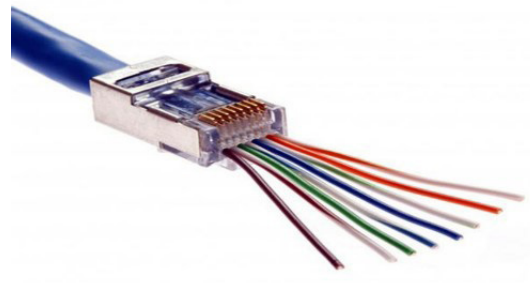


Figure 1.14 RJ-45 Connector

Media Converter: A media converter is a device in computer networks that connects several types of cables such as copper (wired) and fiber optic cables. It operates by converting electrical impulses from copper cables to light signals for fiber optic connections, and vice versa. They are commonly used in schools, offices, and data centers to connect different network systems.

Figure 1.15 Media Converter

Media converters are useful when:

i. The copper cable is not long enough.

ii. Fiber optic cables are needed for faster speed and better performance.

1.4 Networking Devices

The use of computer networks is growing all over the world. So, computer networks are used for the Internet, e-mail, chat, e-banking, etc. The hardware, software, and medium are components of computer network. Network Operating System (NOS) and application software are the software used as network software. The hardware used in the network are computer set, modem, hub, bridge, repeater, switch, router, etc.

Repeater: The repeater is a network connectivity device that accepts weak signals and regenerates them and after that sends the message on its way. It makes it possible for long-distance data transfer. So, it boosts the data signals that are received from the network.

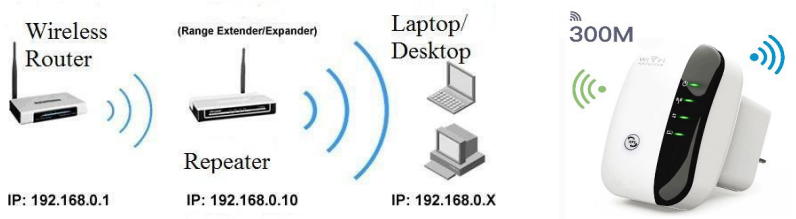


Figure 1.16 Repeater

Hub: Hub is a network connectivity device with multiple ports for connecting different computers on the network. It connects computers in a star topology. It is simple, easier to install and low cost as compared to other devices. It can receive or send information between the computers. Nowadays, hub is replaced by switch.

Figure 1.17 Hub

Switch: The switch is a network connectivity device that joins multiple computers together to share information among them. It coordinates the file server and computers. The selected computer can get information through the switch. It works faster than a hub.

Figure 1.18 Switch

Bridge:

The bridge is a network connectivity device that interconnects two computer networks having similar protocols. It inspects incoming signals and decides whether to forward or discard them. It connects similar networks.

Figure 1.19 Bridge

Figure 1.20 Router

Router:

The router is a network connectivity device that joins multiple wired or wireless networks together. Like a bridge, the router also inspects and forwards data among different networks having similar protocols. It works on the principle of Internet Protocol (IP) address. It is an intelligent device that can determine the best possible path for data transmission. Nowadays, different types of routers are available in the market.

Field Visit Activity 1.1:

Demonstrate and identify network devices and cables.

Field Visit Area

Internet Service Providers (ISPs) / Data Centers / Telecom Companies / IT Companies / Tech Parks / University Network Labs / Broadcasting Stations (TV / Radio), Corporate Office Network Infrastructure / School's computer LAB.

Activity:

i. Identify network devices such as routers, switches, hubs, modems, and access points by name and function.

ii. Recognize and name different types of networking cables, including Ethernet (RJ-45), coaxial, fiber optic, and patch cables.

iii. Differentiate between wired and wireless networking components and understand their roles in a network setup.

1.5 Topology

Network topology is the inter-connected pattern of network components. A network topology may be physical, mapping hardware configuration, or logical, mapping the path that the data must take to travel around the network. A LAN Topology is a cabling structure or pattern of computers in a local area. The four main types of topology are:

i. Bus Topology
ii. Star Topology
iii. Ring Topology
iv. Hybrid Topology

Bus Topology

It is a type of topology that uses a segment of a single cable to connect nodes. All nodes and network devices are connected to a single cable called bus topology. Terminators are attached at the cable's starting and ending points. When it has exactly two endpoints, then it is called linear bus topology.

Features of bus topology

i. It is cost-effective, and the cable required is less compared to other network topologies.
ii. It is used in small networks and is easy to understand.
iii. It is easy to expand networks.
iv. Terminators are necessary at both ends of the cable.

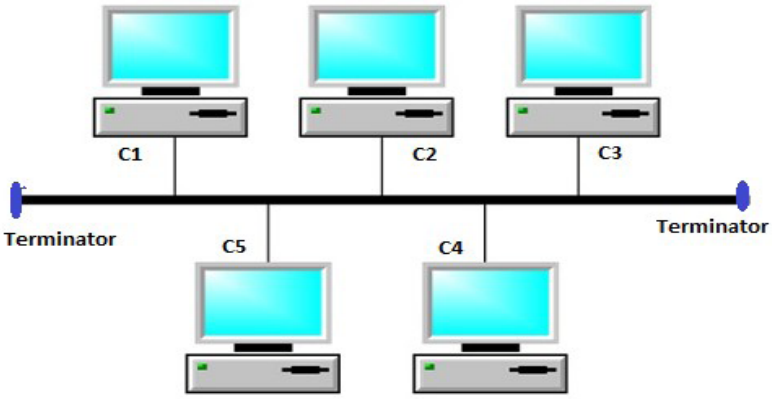


Figure 1.21 Bus Topology

Star Topology

It is a type of topology which connects all nodes to central device called hub/switch through a cable. It is probably the most common topology. Switch acts as a repeater for data flow. Star topology can be used with twisted pair, Optical Fiber and coaxial cable.

Features of star topology

i. It provides fast performance and low network traffic.

ii. It is easy to troubleshoot, set up and modify.

iii. The failure of one node does not affect the rest of the nodes.

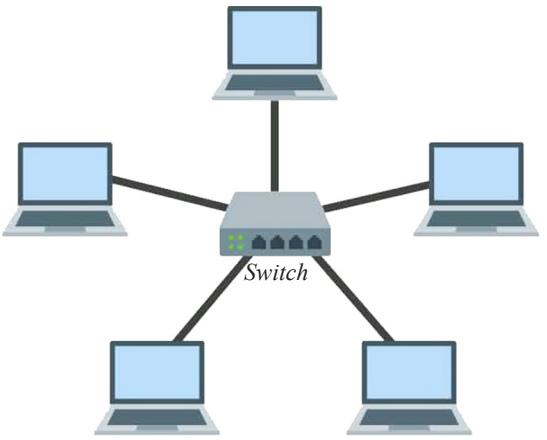


Figure 1.22 Star Topology

Ring Topology

Ring topology forms a closed loop as each computer is connected to another computer with the last one connected to the first which is called ring topology. It uses fiber optic cable to transfer data. Each and every computer has equal responsibilities. A number of repeaters are used for ring topology with large number of nodes. Data is transferred sequentially.

Features of ring topology

i. In this topology, all computers in close loop act as a client or server to transfer the information.

ii. The transmitting network is not affected by high traffic or by adding more nodes.

iii. It is cheap to install or set up and expand.

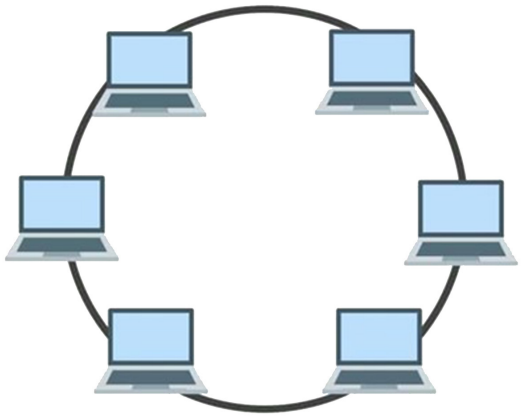


Figure 1.23 Ring Topology

Hybrid Topology

A hybrid topology is a combination of two or more different network topologies such as star, bus, ring, or mesh, within a single network. It allows for greater flexibility and scalability by leveraging the strengths of multiple topologies to meet the specific needs of an organization or network environment. For example, a hybrid topology might consist of interconnected star topologies in different departments of a company, with these star networks linked together using a bus or ring backbone.

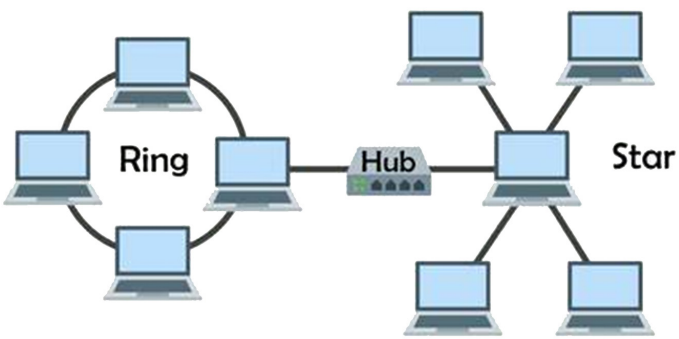


Figure 1.24 Hybrid Topology

Features of hybrid topology

i. Hybrid topology allows adding new devices and technologies as the network grows.

ii. Since it combines different types of networks, it reduces the chances of a complete network failure.

iii. It takes advantage of the best features of different network designs to improve speed and efficiency.

1.6 Overview of Different Networks Based on Coverage

A computer network is a group of two or more computers and devices connected through wire or wireless media to exchange data and information and share hardware, software, and other resources. In other words, the interconnection of many computers to share information is called a computer network. It can share resources like scanners, printers, hard disks, operating system software, application software, etc. within the networking computer. The services provided by the computer networks are print service, message service, application service, file service, database service, etc.

Features of a computer network

i. Network-connected computers can share hardware devices such as scanners, printers, hard disks, etc.

ii. It can communicate and share information all over the world through the Internet.

iii. The computer network supports centralized administration.

iv. Networking also provides the facility of data and software backup systems.

Figure 1.25 Computer Network

Computer networks can be classified according to the area covered. The main four types of computer networks based on geographical location are:

i. PAN (Personal Area Network)

ii. LAN (Local Area Network)

iii. MAN (Metropolitan Area Network)

iv. WAN (Wide Area Network)

PAN (Personal Area Network)

A Personal Area Network (PAN) is a network that is used for connecting devices close to an individual, typically within a range of a few meters. The PANs are designed for personal use and facilitate communication and data sharing between devices such as smartphones, tablets, laptops, wearable devices, and peripherals like keyboards, mice and printers.

Features of PAN

i. PANs work within a small area, usually up to 10 meters.

ii. PANs link devices like smartphones, tablets, laptops, smartwatches, and fitness trackers.

iii. Users can transfer files, photos, videos, and other data between connected devices.

iv. Setting up a PAN is easy and requires only basic configurations.

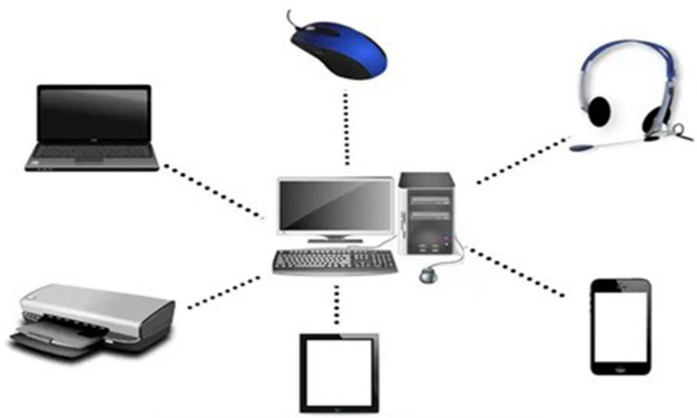


Figure 1.26 PAN

Local Area Network (LAN)

Local Area Network (LAN) is a network limited to small areas like a room, a building, a school, a college, etc. It is generally connected through wire media. It can cover up to one kilometer for transferring data and information between the computers. It has high speed data transfer capacity than the MAN and the WAN. A computer in the LAN can be connected through wireless media and is referred to as the WLAN (Wireless Local Area Network). The network in school's computer lab is the example of the LAN.

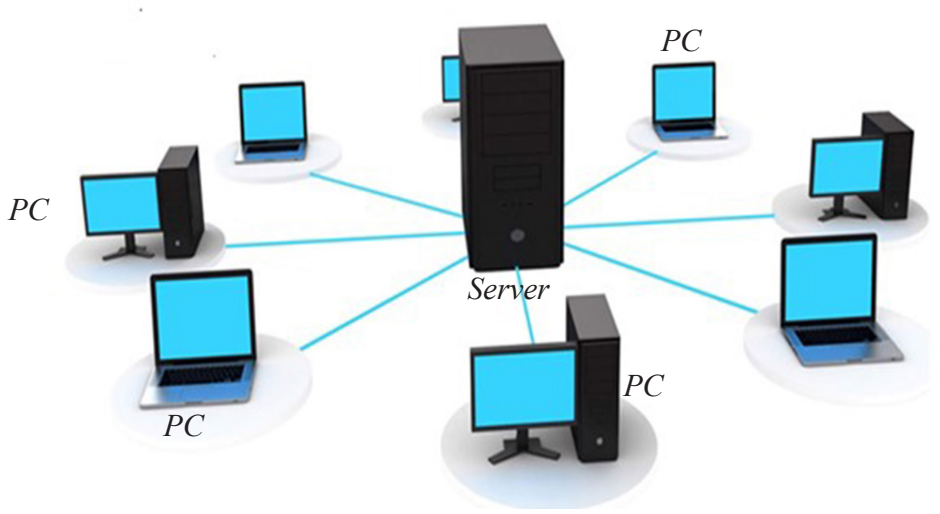


Figure 1.27 LAN

Features of Local Area Network

i. The coverage area is generally a small geographical area such as a room, building, and a few kilometres.

ii. A LAN connects multiple devices to share a transmission medium.

iii. LANs commonly use topologies such as BUS and RING.

iv. The communication quality is better in the LAN, the transmission error rate is low as compared to the WAN.

MAN (Metropolitan Area Network)

A Metropolitan Area Network (MAN) is a network within an area like a valley, city, metropolitan, etc. connected through wire or wireless technology or media. It covers more area than the LAN and less area than the WAN. It has a higher data transfer capacity than the WAN and is slower than the LAN. It is used in medium-sized organizations having various branches within a city or valley. Cable TV networks and Telephone networks providing high-speed DSL lines are the examples of the MAN.

Features of Metropolitan Area Network

i. The network size ranges from a few buildings on a campus to an entire city.

ii. It is normally owned by single or multiple organizations.

iii. It facilitates the sharing of regional resources.

iv. They provide uplinks for connecting the LANs to the WANs and the Internet.

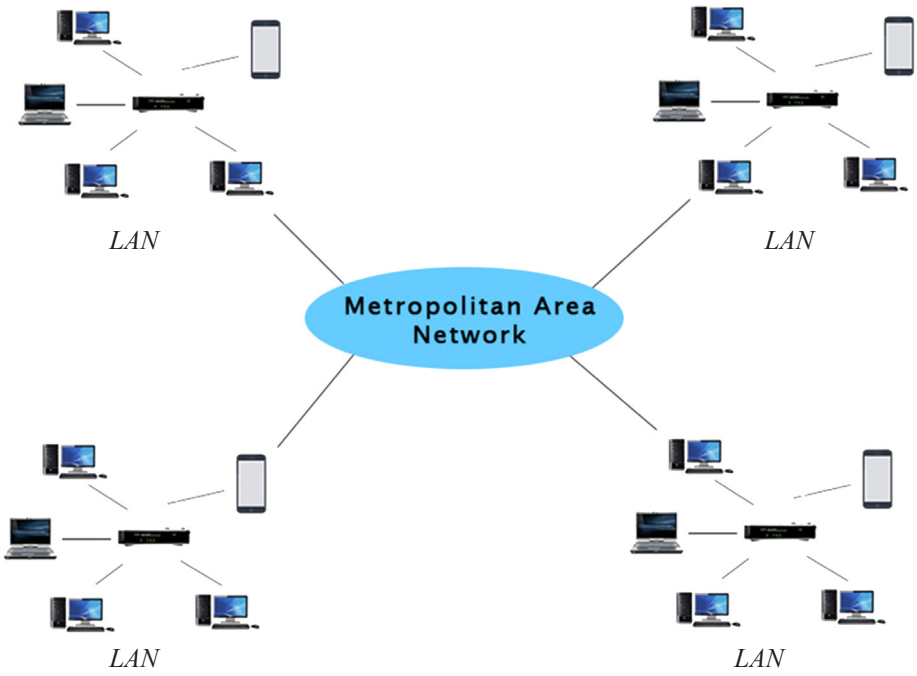


Figure 1.28 MAN

Wide Area Network (WAN)

A Wide Area Network is a network that is extended to a large geographical area in which computers are connected through wireless technology or media as satellite communication. It is the largest computer network. Internet, 4G Mobile Broadband Systems, and satellite communication are the best examples of the WAN.

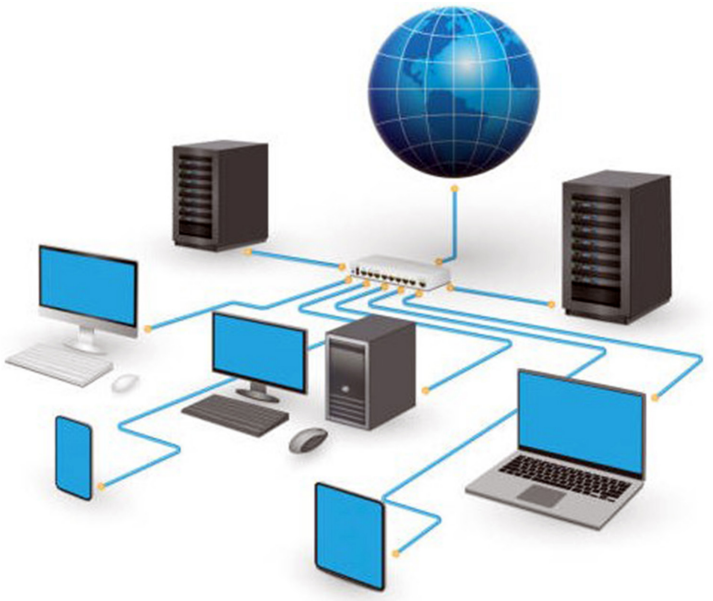


Figure 1.29 WAN

Features of Wide Area Network

i. It covers the whole world and cannot be restricted to any geographical location.

ii. It is normally owned by multiple organizations.

iii. Communication links are provided by public carriers like telephone networks, satellites, network providers, cable systems, etc.

They have low data transfer rates and high propagation delay, resulting in low communication speed.

Fun Facts

The first computer network, known as ARPANET, was established in 1969 and served as the precursor to the internet. The first message transmitted was expected to be "LOGIN"—but the system failed after "LO"!

1.7 Network Architecture

Network architecture defines how the computer communicates and interacts with each other on the network. It is the combined form of topologies and data transmission mode between the different computers in the network. The two types of network architecture are:

Client-Server Network

A network model in which one or more powerful computers (servers) provide various network services, and other computers (clients) access those services to perform user tasks, is known as a client-server network. In such network, there exists a central controller which is called a server. A server is a specialized computer that controls the network resources and provides services to other computers in the network. Microsoft Windows Server, Linux (such as Ubuntu Server or CentOS), and UNIX are common examples of operating systems of client-server networks.

Features of client-server

i. The centralized backup is possible in a client-server computer network.

ii. The use of dedicated servers also increases the speed of sharing resources.

iii. Security is better in this network as all the shared resources are centrally administered.

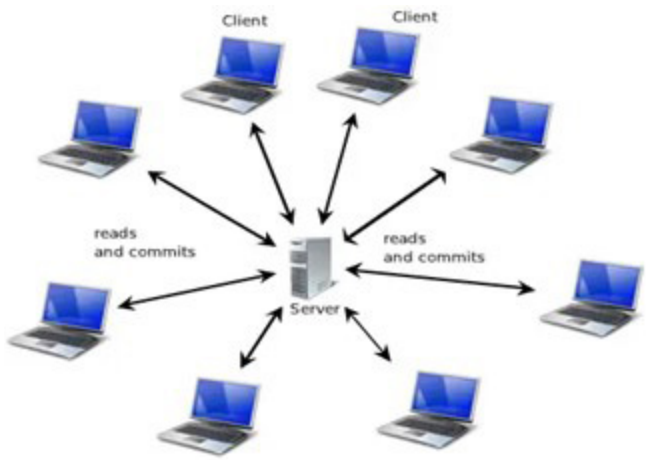


Figure 1.30 Client-Server

Peer-to-Peer Network

In this type of network, all nodes on the network have equal relationships with each other. It is also called a work group. It can share data, hardware, and software. It is suitable for small areas such as small offices, small rooms, buildings, etc. It uses many operating systems such as Windows 11, macOS Sonoma, and various Linux distributions like Ubuntu 24.04 and Fedora 40, etc., that have built-in features to support peer-to-peer networks.

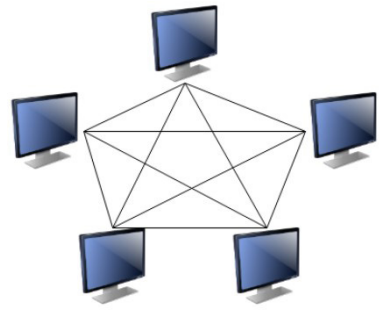


Figure 1.31 Peer-to-Peer Networks

Features of Peer-to-Peer Network

i. All resources and contentions can be shared in this network.

ii. System administrator is not required and each user can control their shared resources.

iii. One computer's failure doesn't affect other peers' functioning.

Network Protocols

A set of rules followed for interconnection and communication between computers in a network is called a protocol. So, it is the rules followed while sending and receiving information using network software. NCP (Network Control Protocol) was the first protocol. Some common protocols are as follows:

i. TCP/IP (Transmission Control Protocol/Internet Protocol) is used on the Internet.

ii. HTTP (Hyper Text Transfer Protocol) is used to transfer HTML documents in WWW.

iii. HTTPS (Hypertext Transfer Protocol Secure) is a secure way to send and receive information on websites, protecting your data from hackers.

iv. DHCP (Dynamic Host Configuration Protocol) is a system that automatically gives devices on a network an IP address so they can connect to the internet.

v. SMTP (Simple Mail Transfer Protocol) is used in sending e-mail.

vi. FTP (File Transfer Protocol) is used to transmit files between computers.

1.8 Concept of IP Addressing

Internet Protocol (IP) is responsible for giving each device on a network a unique address and routing data between devices. IP ensures that data sent from one computer reaches the correct destination. IPv4 (Internet Protocol version 4) and IPv6 (Internet Protocol version 6) are two different versions of the Internet Protocol.

IPV4 (Internet Protocol version 4)

IPv4 is the fourth version of the Internet Protocol and the most widely used version on the Internet today. It uses 32-bit addresses, allowing for approximately 4.3 billion unique addresses. IPv4 addresses are expressed in dotted-decimal notation, with each address consisting of four numbers separated by periods (e.g., 192.168.1.1). The rapid growth of Internet-connected devices has led to IPv4 address exhaustion, as the number of available addresses is limited. To address this limitation, techniques such as Network Address Translation (NAT) and private IP address ranges are used to conserve IPv4 addresses.

IPV6 (Internet Protocol version 6)

IPv6 is the sixth version of the Internet Protocol and was developed to address the limitations of IPv4, particularly address exhaustion. It uses 128-bit addresses, providing a vastly larger address space compared to IPv4. IPv6 addresses are expressed in hexadecimal notation, with each address consisting of eight groups of four hexadecimal digits separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a 2e:0370:7334). IPv6 includes features such as stateless address auto-configuration, built-in security features, and support for multicast communication. While IPv6 adoption has been slower than anticipated, it is gradually gaining traction as the Internet transitions to accommodate the growing number of connected devices and services.

Comparison IPv4 and IPv6

IPv4	IPv6
IPv4 has 32-bit address length.	IPv6 has 128-bit address length.
It supports manual and DHCP address configuration.	It supports auto and renumbering address configuration.
In IPv4 end to end connection integrity is unachievable.	In IPv6 end to end connection integrity is achievable.
It can generate 4.29×10^{9} address space.	It can generate 3.4×10^{38} address space.
Address representation of IPv4 is in decimal.	Address representation of IPv6 is in hexadecimal.
Encryption and authentication facility is not provided.	Encryption and authentication are provided.
It has header of 20-60 bytes.	It has header of 40 bytes fixed.

Activity 1.2:

"Check IP Address and Default Gateway"

i. Identify the concept of an IP address and its role in a computer network.
ii. Check the IP address and default gateway of a device.
iii. Understand the difference between private and public IP addresses and why a default gateway is important in network communication.
iv. Use command-line tools (such as ipconfig on Windows) to view network configurations and identify the device's IP address and default gateway.
v. Interpret the information displayed, such as:
   - o IPv4 vs. IPv6 address
   - o Subnet mask
   - o Default gateway (router address)

1.9 Concept of the Internet, Intranet, and Extranet

Internet is a global network of computers and devices that allows people to share information and communicate across long distances. It works using the TCP/IP protocols, which help data travel between computers using wired or wireless connections. The Internet provides many services like web browsing, e-mail, social media, online shopping, video streaming, and gaming. It has changed how we communicate, work, and learn.

Figure 1.32 Internet

An **Intranet** is a private network used within an organization to help employees communicate, share information, and work together. Unlike the Internet, which is accessible to everyone, an intranet is only accessible to authorized users within the organization. Intranets use Internet technologies like TCP/IP, HTTP, and HTML to provide services such as email, file sharing, internal websites, and online collaboration tools. The main goal of an intranet is to improve communication, boost productivity, and make it easier to access company resources.

An **Extranet** is a private network that allows an organization to share information with selected external users like customers, suppliers, or business partners. It works like an intranet, but instead of being limited to employees, it provides controlled access to people outside the organization. It uses Internet technologies to enable secure communication and collaboration between internal and external users. To keep information safe, access is restricted to authorized users only.

Comparison among Internet, Intranet, and Extranet

S.No	Internet	Intranet	Extranet
1	The Internet connects millions of private, public, academic, business, and government networks.	An intranet is a private network accessible only to an organization's staff.	Extranet is a controlled private network allowing access to outsiders, typically partners and clients.
2	Public access that anyone can connect.	Restricted to employees of the organization only.	Limited access that allows external users, like partners or clients, can connect.
3	The main purpose is to share information and connect people worldwide.	The main purpose is to share information and resources within an organization.	The main purpose is to share specific information between organizations and their partners.
4	Generally less secure, which requires precautions (like firewalls) for safety.	More secure and restricted to users within the organization.	Secure, but needs to balance accessibility for outsiders.
5	The examples include websites, social media, and online services like Google and Facebook.	The examples include company portals, internal communication platforms.	The examples include vendor portals, client access systems for specific resources.

Activity 1.3:

Demonstrate the use of the following commands:
ping, ipconfig, tracert, nslookup

Activity:

i. Understand the purpose of basic network commands used in troubleshooting and analysing network connections.

ii. Use the ping command to test if a device (like a website or another computer) is reachable over the network.

iii. Use the ipconfig command to check the computer's IP address, subnet mask, and default gateway.

iv. Use the tracert command to trace the path that data takes from the local computer to a destination on the Internet, observing the number of hops.

v. Use the nslookup command to find the IP address of a domain name (e.g., finding the IP address for www.google.com).

Activity 1.4:

Demonstrate RJ45 and Fiber connectors.

i. Identify and describe the physical appearance and purpose of an RJ45 connector used in Ethernet cables.

ii. Identify and describe fiber optic connectors (such as SC, LC, or ST types) used in high-speed data transmission.

iii. Understand the difference between RJ45 (used for electrical signals in copper cables) and fiber connectors (used for light signals in optical cables).

iv. Observe and explain how RJ45 connectors are crimped onto twisted-pair cables and how they plug into network ports.

v. Understand the advantages of fiber optics, such as faster speeds and longer-distance communication compared to copper cables.

vi. Learn the typical uses of RJ45 in local area networks (LANs) and fiber connectors in backbone or internet service provider (ISP) networks.

Exercise

1. Write the full forms of the following abbreviations:

a) DSL b) bps c) LAN

d) TCP/IP e) IPv6 f) ISP

g) RFID h) CAT6 i) NCP

j) DNS

2. Choose the correct answer from the given options:

i. Which of the following is a broadband Internet connection?

a) DSL b) Fiber optic

c) Cable internet d) All of the above

ii. What is throughput?

a) Theoretical speed of a network

b) Actual data transferred in a given time

c) Length of a network cable

d) Number of users

iii. What is a data packet?

a) A physical network device

b) A unit of data sent over a network

c) A type of wireless method

d) A security tool

iv. Which of the following is a type of bounded (guided) media?

a) Fiber optic b) Infrared

c) Microwave d) Laser

v. **Which term refers to sending data from Earth to a satellite?**

a) Downlink b) Modulate

c) Uplink d) Download

vi. **What is the RJ45 connector mainly used for?**

a) USB connections b) Telephone lines

c) Ethernet networking d) Fiber optics

vii. **What is the connection pattern of computers in a network called?**

a) Protocol b) Topology

c) Twisted pair d) Structure

viii. **Which topology uses a hub to connect all devices?**

a) Ring topology b) Bus topology

c) Star topology d) Hybrid topology

ix. **What type of network connects LANs over large areas?**

a) PAN b) MAN

c) WAN d) CAN

x. **Which of the following are Internet services?**

a) IRC b) Telnet

c) Email d) All of the above

xi. **Which protocol is used to transfer files between computers?**

a) FAQ b) IRC

c) FTP d) TPF

xii. **What is the length of an IPv4 address?**

a) 16 bits b) 32 bits

c) 64 bits d) 128 bits

xiii. What does an IP address identify?

a) A software

b) A network cable

c) A specific device on the network

d) A computer brand

xiv. Which protocol is commonly used for sending emails?

a) HTTP b) FTP

c) SMTP d) DHCP

xv. Which device strengthens weak network signals for long distances?

a) Switch b) Router

c) Repeater d) Bridge

3. Write short answers to these questions.

a) What is broadband? How is it different from dial-up connections?

b) Define bandwidth. How is it measured?

c) What is a data packet in networking?

d) What is frequency in telecommunications?

e) What is the function of a repeater?

f) What is a computer network? How is it useful?

g) Why is wireless communication becoming more popular today?

h) Describe the RJ45 connector. Where is it commonly used?

i) What is a media converter? Mention its main function.

j) What is the difference between bandwidth and throughput?

k) How does Wi-Fi transmit data without cables?

l) How does data travel from one computer to another in a network?

m) How does data flow in a ring topology?

n) Mention one real-life use of satellite communication.

o) List two types of communication media and give one example of each.

4. Write long answers to these questions.

i. What is communication media?

ii. Differentiate between LAN and MAN.

iii. Explain the differences between client-server and peer-to-peer network architectures.

iv. Suppose your school wants to set up a network in three separate buildings. What type of network should be used? Justify your answer by explaining how it would support communication, file sharing, and internet access.

v. Create a network model for your home that includes three PCs, one printer, and one mobile device connected to the internet. Describe the devices, connection types (wired or wireless), and technologies (router, switch or Wi-Fi) you would use.

vi. Which network type would be more suitable for a small office: client-server or peer-to-peer? Justify your answer by comparing features like cost, security, scalability, and management.

vii. Design a simple layout for a school computer lab network using at least one switch, 10 computers, and internet access. Explain your design choices and how the devices would communicate with each other.

Case Studies for Classroom Discussion

Case Study 1: Choosing the right topology

A school is preparing to modernize its computer lab. The network must be stable and easy to maintain. The lab includes 36 computers, all of which should be connected to the Internet and capable of instantly sharing files.

- **Task**: Recommend a suitable network topology and justify your answer.
- **Focus concepts**: Star topology, switch, Ethernet cables, LAN

Case Study 2: Wi-Fi connectivity issue

A student complains that his home Wi-Fi works properly on the ground floor but is very slow on the first floor.

- **Task**: Identify possible reasons and suggest solutions.
- **Focus concepts**: Wi-Fi, bandwidth, signal interference, repeater, router placement

Case Study 3: Office network upgrading

An office in Birgunj wishes to upgrade from 3G to 5G for faster communication and improved cloud service access. They are now using obsolete networking devices.

- **Task**: Suggest upgrades and explain the benefits of moving to 5G.
- **Focus concepts**: 3G vs. 4G vs. 5G, throughput, latency, IoT, mobile broadband

Unit 2 Database Management System

Fig. 2.0

2.1 Introduction to database

Let's begin with a thought.

Have you ever visited a library and noticed how the librarian finds your book in seconds, even among thousands of books?

a. How do schools keep track of students' names, grades, fees, and attendance?
b. Where do online shopping sites store details about thousands of products?

All these places **store, manage, and retrieve** a lot of information. But how do they do it so efficiently?

*A **database** is a systematic way of storing and managing data so that it can be used easily and efficiently*. In our daily lives, we come across many situations where we need to store and retrieve information. For example, schools keep records of students' names, attendance, exam marks, and addresses. Similarly, mobile phones save contacts, messages, and call history. All these are examples of databases, where information is arranged properly so that it can be accessed whenever needed.

Databases help us to store large amounts of information without confusion. They allow us to search for specific data quickly, add new information, update existing details, or even remove old records. This makes them very useful in places like hospitals, banks, online shopping websites, offices, and schools. A database works like a digital cupboard, it keeps everything in order and safe from getting lost or mixed up.

In today's digital world, databases play an essential role in most computer systems. Whether we are booking a flight ticket, logging into a social media account, or watching videos on a streaming app, databases are working in the background to manage the data. Learning how databases work is the first step toward understanding how modern technology stores and handles information in a smart way.

Let's think !

Why might a school want to store student records in a better way than a paper file or Excel sheet?

These days world runs on data. Here's how databases are used around us:

Area	How databases are used
Social media	stores user profiles, posts, comments
Online shopping	keeps product info, prices, orders, user reviews
Banking	manages accounts, transactions, balances
Healthcare	stores patient records and medical histories
Cloud databases	companies like Google and Amazon use online databases that can be accessed from anywhere

Cloud databases are becoming popular. You do not need a physical server they work over the Internet and grow with your needs.

2.2 Data, Database, Database Management System

Data is a collection of raw facts, figures, or information that may not have meaning on its own. It can be in the form of numbers, words, images, or symbols. For example, your name, age, mobile number, or marks in exams are all types of data. When data is organized or processed in a useful way, it becomes information. In computers, data is the basic input that is stored, processed, and used to make decisions or perform tasks.

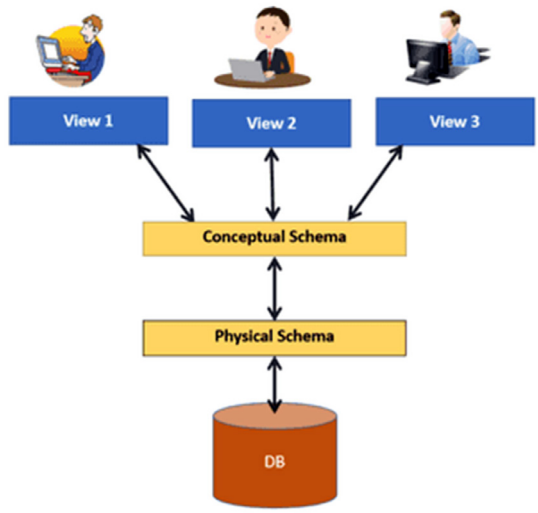


Figure: 2.1 Database

Information is the useful result we get after arranging or processing data. It helps us understand things better and make decisions.

Example:
If "80, 85, 90" are marks (data), then
"The student scored an average of 85 marks" is information.

Differentiate between data and information

Aspect	Data	Information
Definition	raw, unprocessed facts or figure	processed and organized data with meaning.
Form	can be numbers, text, symbols, images etc.	data that is analyzed and structured.
Meaning	no clear meaning on its own.	meaningful and is useful for understanding or decision - making.
Example	"2082, 95, Binay"	"Binay scored 95 marks in 2082 BS, and is one of the top performers."

A **database** is an organized collection of data that can be easily accessed, managed, and updated. It stores information in tables, making it easier to find and use.

For example, in a school database, student details like names, grades, and attendance are stored. Databases help businesses, schools, and websites keep large amounts of data safe and easy to retrieve when needed.

Before the influence of Information Technology, data was maintained using traditional methods. However, in today's world, most databases are maintained using computerized systems. A **traditional file system** refers to a system where data is stored and managed manually, often using physical storage methods like paper records or file cabinets. In contrast, a **computerized database** uses modern digital technology to store, retrieve, and manage data, offering faster and more efficient operations.

Group Activity

Divide into small groups. Each group should:

i. collect data from 5 classmates (name, age, favourite subject).

ii. organize this into a small table (like a database).

iii. discuss how a computer program such as DBMS could make this process easier if there were 100+ students.

Database Management System (DBMS)

Let's begin with a thought.

Imagine you work for a company that handles thousands of customer orders each day.

i. You need to store customers' names, addresses, products ordered, and payment details.

ii. You also need to retrieve a customer's order history whenever they ask.

iii. You want to update an order if a customer changes their mind.

Would you want to store this in paper files? Or would you prefer to use a software system to manage all this information?

The answer is clear: A Database Management System (DBMS)!

A **Database Management System (DBMS)** is an application software designed to store, manage, and organize data efficiently. It allows us to create, retrieve, update, and delete data in a structured way, ensuring data integrity, security, and ease of access. A DBMS acts as an intermediary between users and the database, making data management seamless and secure.

Here are some common examples of DBMS software:

a. MySQL - Open-source and widely used in web applications

b. Oracle Database - Powerful commercial DBMS used in large enterprises

c. Microsoft SQL Server - A popular DBMS from Microsoft, used in business environments

d. PostgreSQL - An open-source DBMS known for its advanced features and reliability

e. MongoDB - A NoSQL database used for handling large volumes of unstructured data

f. SQLite - Lightweight and used in mobile apps and embedded systems

Recent Trends and Technologies in DBMS:

1. **Cloud database:** With the rise of cloud computing, databases are increasingly hosted on the cloud, offering scalability, flexibility, and cost-efficiency.

2. **NoSQL database**: Non-relational databases like MongoDB and Cassandra have gained popularity due to their ability to handle large volumes of unstructured data, making them ideal for big data applications.

3. **In-Memory database:** Technologies like Redis (**Re**mote **Di**ctionary **S**erver) and HANA (**H**igh Performance **An**alytic **A**ppliance) store data in the system's main memory rather than on disk, enabling ultra-fast data retrieval and processing.

4. **Distributed database:** These databases spread data across multiple servers or locations, enhancing performance, fault tolerance, and availability.

Features of DBMS

1. **Data storage:** A DBMS stores data in tables (rows and columns) where each piece of information has a specific place.
2. **Data retrieval:** It allows you to quickly retrieve data using simple commands.
3. **Data manipulation:** It supports actions like adding, deleting, and updating data.
4. **Data integrity:** It ensures that the data is consistent and follows certain rules. For example, it won't allow an incorrect phone number format in a database.
5. **Data redundancy control:** DBMS minimizes data duplication by centralizing data storage and promoting normalization, ensuring that data is not repeated unnecessarily.
6. **Security:** The DBMS controls who can access the data and what actions they can perform on it, making sure sensitive information is protected.
7. **Data backup and recovery:** It ensures that your data is safe and can be recovered in case of any accidental loss.

Let's think:

Why do you think it's necessary to use a DBMS rather than just using a simple spreadsheet or text file for storing large amounts of data?

Benefits of DBM

i. Easy data storage and retrieval

ii. Reduces data redundancy

iii. Ensures data consistency and accuracy

iv. Allows multiple user access

v. Supports data backup and recovery

Challenges of DBMS

i. Requires technical knowledge to manage

ii. Initial setup cost can be high.

iii. Data security risks if not properly managed.

iv. Complex systems may need regular maintenance.

v. Performance may be slow with very large data.

Relational Database Management System (RDBMS): Relational Database Management System is a type of DBMS that stores data in tables made up of rows and columns. Each table is linked to others through common fields, which makes it easy to organize and manage related information. Since you are learning MySQL, it's important to know that MySQL is a popular example of an RDBMS. Other examples include Oracle, PostgreSQL, and Microsoft SQL Server. RDBMS is widely used today in websites, apps, banking systems, schools, and businesses to handle large amounts of data in an organized and secure way.

2.3 Data Types

Let's Think.

i. What might go wrong if we try to store a phone number as a date?

ii. Why is it useful to separate numbers from text when designing a database?

In SQL, data types define the kind of data that can be stored in each column of a database table. For example, numbers like age or ID use numeric data types such as INT or DECIMAL, while names and addresses use text data types like VARCHAR. Dates, such as a birthdate or event time, use date/time data types like DATE or DATETIME. Choosing the right data type helps ensure the data is stored correctly and efficiently.

Here are some of the most commonly used SQL data types, grouped by the kind of data they store:

1. Numeric Data Types

Used for storing numbers.

Data types	Description
INT	Whole numbers.
DECIMAL(p,s)	Exact numbers with decimal points. p is the total number of digits, s is the number of digits after the decimal
FLOAT	Approximate numbers with decimals (less precise but can store very large or small numbers)
SMALLINT	Smaller range of whole numbers
BIGINT	Very large whole numbers

2. Character (Text) Data Types

Used for storing letters, words, or text.

Data types	Description
CHAR(n)	Fixed-length text. Always stores exactly *n* characters.
VARCHAR(n)	Variable-length text up to *n* characters. More flexible.
TEXT	Long blocks of text. Used for storing large paragraphs.

3. Date and Time Data Types

Used for storing dates and times.

Data types	Description
DATE	Stores only the date (e.g., 2047-07-06)
TIME	Stores only the time (e.g., 14:30:00)
DATETIME	Stores both date and time together.
TIMESTAMP	Similar to DATETIME, often used to record events automatically

4. Boolean Data Type

Used for storing true/false values.

Data Type	Description
BOOLEAN	Stores either TRUE or FALSE

2.4 Table

A table in a database is a structured way to store data in rows and columns, much like a spreadsheet. Each table represents a specific type of information, such as students, books, or products. The columns define the types of data (like name, age, or price), while each row holds a single record or entry. Tables help organize data clearly, making it easy to search, sort, and manage large amounts of information efficiently. Table is also known as the primary building block of a database.

Here are the key features of table:

a. **Rows and Columns:** Data is arranged in horizontal rows and vertical columns.

b. **Unique rows:** Each row stores one complete record and is often identified by a unique ID (called a primary key).

c. **Defined schema:** Each column has a specific data type like text, number, or date.

d. **Organized data:** Tables help keep data neat, clear, and easy to understand.

e. **Easy to search and update:** We can quickly search, add, update, or delete records from a table.

f. **Relationships:** Tables can be linked to other tables using keys, making data more connected and meaningful.

Field is also known as a **Column**, contains a single type of data under a specific category. It is a part of a database table that holds one piece of information about an entity. Each field stores data of a specific type, such as text, numbers, dates, or other forms of information.

For example, in a student database, fields might include **Name**, **Roll Number**, **Class**, and **Date of Birth**. Each of these fields stores a specific type of data for all students in the database.

Student_ID	Name	Class	Date of Birth
101	Aarosh Pokharel	10	2010-03-12
102	Subigya Nepal	10	2011-07-25
103	Aarambha Gharti	9	2013-11-10

In this table:

a. Student_ID, Name, Class and DateOfBirth are the Fields.

b. Each field holds a specific type of data: Student ID holds numbers, Name holds text, Age holds numbers and Date of Birth holds dates.

Key points about Fields:

i. A field is a vertical column in a database table.

ii. Each field holds a specific type of data for every record in the table.

iii. The field's name describes the type of data stored, such as Name, Age, or Date of Birth.

iv. The data type of a field determines the kind of data it can store (e.g., text, numbers, dates).

Records: A record, also known as a row, represents a single, complete set of data about one entity in a database table. Each record contains information for each field (column) in the table. In simple terms, a record is like a "complete entry" for one item, such as a student, employee, or product.

For example, in a student database, a record would represent the information about one student, with details for each field like Name, Roll Number, Class, and Date of Birth.

S_ID	Name	Class	Date of Birth
101	Binay Adhikari	10	2010

In this table:

a. The entire row (the one with 101, Binay Adhikari, 10, 2010) is one record.

b. It contains the complete information about one student, Binay Adhikari

Key points about records:

i. A record represents all the information about one entity (like one student, one employee, or one product)

ii. Each record in a table is made up of fields, and every field holds one specific type of data.

iii. Records are organized horizontally in a table, where each row contains the complete data for one entry.

2.5 Keys: Primary Key and Foreign Key

In a database, keys are special fields (columns) used to uniquely identify data and link tables together. They help keep the data organized, accurate, and connected.

1. **Primary Key:** A primary key is a field (or combination of fields) that uniquely identifies each record in a table. No two rows can have the same value in the primary key column, and it cannot be empty (NULL).

   Features of Primary key

   a. Uniquely identifies each record
   b. Cannot contain duplicate values
   c. Cannot be NULL
   d. Only One primary key is allowed per table

Example:

StudentID	Name	Class	Age
101	Biwash	10	15
102	Bidhya	10	14
103	Biraj	9	14

Here, StudentID is the primary key because it uniquely identifies each student.

2. **Foreign Key**: A Foreign Key is a field in one table that refers to the primary key in another table. It creates a relationship between two tables and helps ensure that the linked data is valid. Foreign Key is also known as a linking pin between two tables.

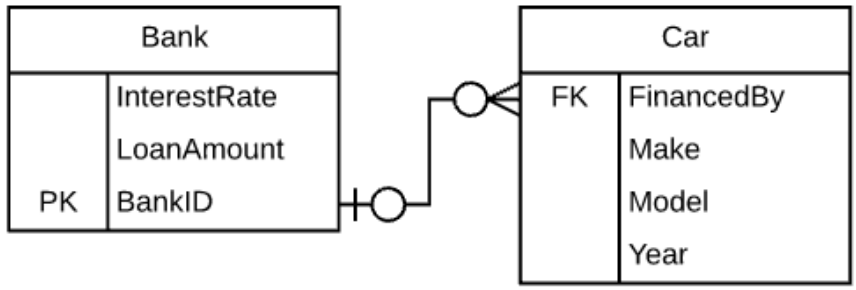


Figure : 2.2 Relational Table

Above diagram shows the relationship between a Bank and a Car. In the Car table, FinancedBy is a foreign key (FK) that connects to the BankID (Primary Key) in the Bank table.

Features of Foreign Key

a. Creates a relationship between tables.

b. Refers Primary Key of another table.

c. Enforce relationships.

d. Ensure data integrity.

e. Consist in one or more columns.

Example:

Table: Students

StudentID	Name	Class
201	Nidhi	10
202	Sampada	9

Table: Library

BookID	BookTitle	StudentID (Foreign Key)
B_01	Nepali History	201
B_01	Database Basics	201
B_03	Learn Python	202

- In the Library table, StudentID is a Foreign Key that refers to the StudentID in the Students table.
- This ensures that only valid student IDs can be entered in the Library table.

3. **Composite Key**: A Composite Key is made by combining two or more fields to uniquely identify a record. This is used when a single field alone cannot ensure uniqueness.

Example:

StudentID	Subject	Marks
301	Nepali	85
301	Social	92
302	Computer	95
303	Social	78

a. Here, neither StudentID nor Subject alone can uniquely identify a row.

b. But the combination of **StudentID + Subject** can - making it a Composite Key.

Practise Yourself:

a. Which of the following can be a primary key?

i. Student Name ii. Date of Birth
iii. Student ID iv. Class Name

b. What is the purpose of a foreign key?

i. To store numbers only ii. To uniquely identify each record
iii. To link one table with another iv. To create duplicate values

c. Which key is formed by combining two or more fields?

i. Foreign Key ii. Primary Key
iii. Composite Key iv. Super Key

Types of Relationship

Database relationships are fundamental concepts in relational database design. They allow data to be stored efficiently, reduce redundancy, improve data integrity, and enable accurate retrieval of information. In a database, a relationship is a logical connection between two or more tables, typically based on a common field such as a key. These relationships make it possible to link data stored in separate tables and access them together in a meaningful and organized way. There are three types of relationship in database they are:

1. One-to-one (1 : 1)
2. One-to-many (1 : M)
3. Many-to-many (M : M)

1. **One-to-one relationship**: In a one-to-one relationship, each record in one table is connected to only one record in another table. This type of relationship is not very common, but it is useful when we want to separate closely related information into two different tables.

   Example: A person can have only one passport, and each passport is issued to only one person.

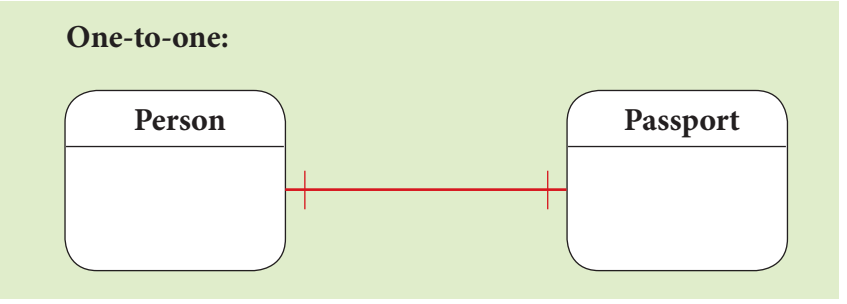


Figure: 2.3 (1:1 Relationship)

2. **One-to-many relationship:** In a one-to-many relationship, a single record in one table is linked to multiple records in another table. However, each record in the second table is connected to only one record in the first table. This is the most common type of relationship in databases. It is useful when one item is related to several items of another type.

e.g. A teacher may teach many classes, but each class is taught by only one teacher.

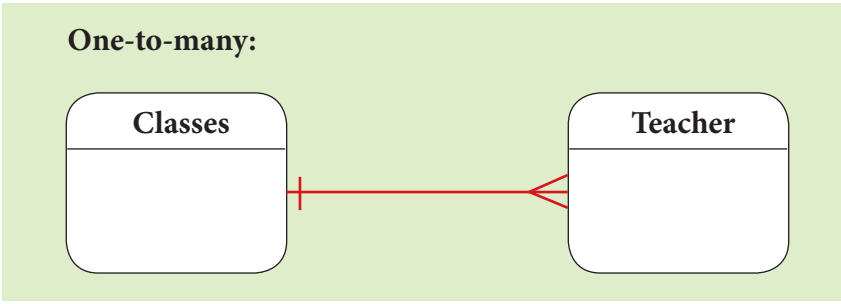


Figure: 2.4 (1:M Relationship)

3. **Many-to-many relationship**: In a many-to-many relationship, multiple records in one table can be related to multiple records in another table. This type of relationship requires a third table, called a junction table, to link the records from both tables.

   e.g. A student can enrol in many courses, and each course can have many students.

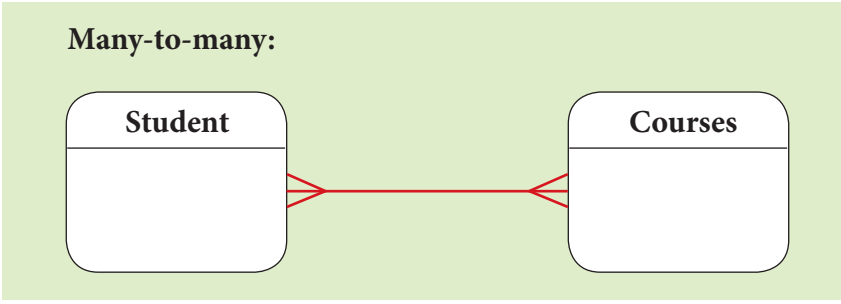


Figure: 2.5 (M:M Relationship)

2.6 Introduction to MySQL:

In today's digital world, data is everywhere — from school records to online shopping and social media. To manage this data, powerful tools are needed. One of the most popular tools for working with data is MySQL, an open-source database management system used all over the world.

MySQL is a popular relational database management system (RDBMS) that uses SQL (Structured Query Language) to store, manage, and organize data. Originally

developed by a Swedish company called MySQL AB, it is now maintained by Oracle Corporation. MySQL is widely used in web development, software applications, and data-driven systems due to its speed, reliability, and free availability.

Figure: 2.6

It allows users to interact with databases using SQL queries without needing to understand the full internal structure of the database. SQL provides statements for defining, manipulating, and controlling data. A query is a command sent to the database to retrieve, insert, update, or delete data.

Today, MySQL is commonly used in websites, schools, banks, offices, and mobile applications. Learning its basics helps students understand how data is stored, searched, and managed in real-world systems.

Features of MySQL

i. Simple language

ii. Data retrieval

iii. Data manipulation

iv. Data definition

v. Multiple users

vi. Security features

vii. Used in modern applications

Download MySQL

MySQL is a widely-used, open-source relational database management system (RDBMS) that allows users to store, manage, and retrieve data efficiently. Downloading the latest version, MySQL 8.3, ensures access to the newest features and security updates. This guide will walk you through the process of downloading MySQL 8.3 on a Windows system.

LAB Activity

Step 1: Visit the Official MySql Download Page: **https://www.mysql.com/downloads/**

Step 2: Click on MySQL Community (GPL) Downloads:

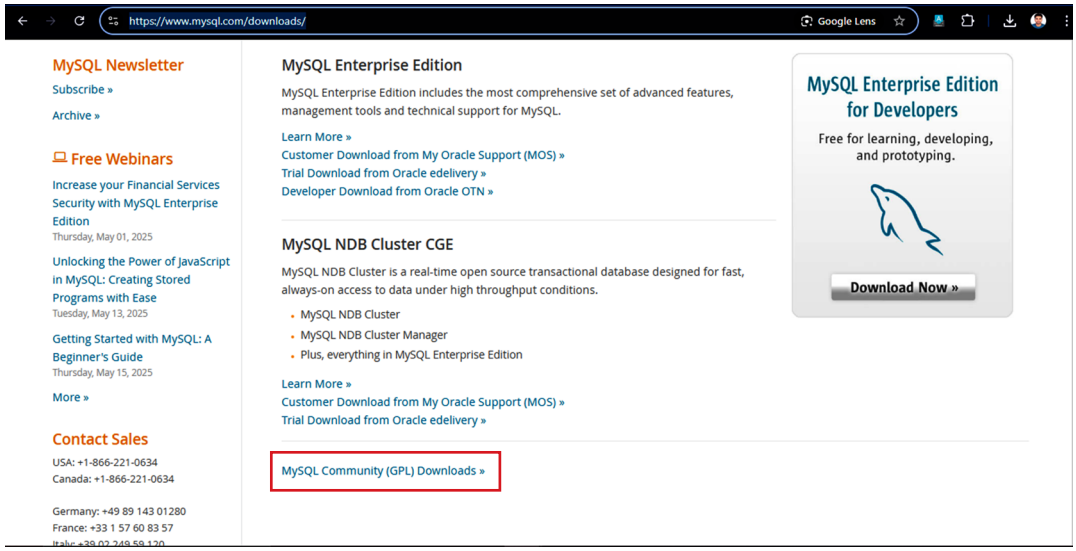


Figure: 2.7

Step 3: Click on MySQL installer for windows from MySQL:

Figure: 2.8

Step 4: After performing the above operation on MySQL Installer for windows from MySQL, click on the second option Windows (x86, 32-bit), MSI Installer. [Note : you can change version and operating system as per your requirement]

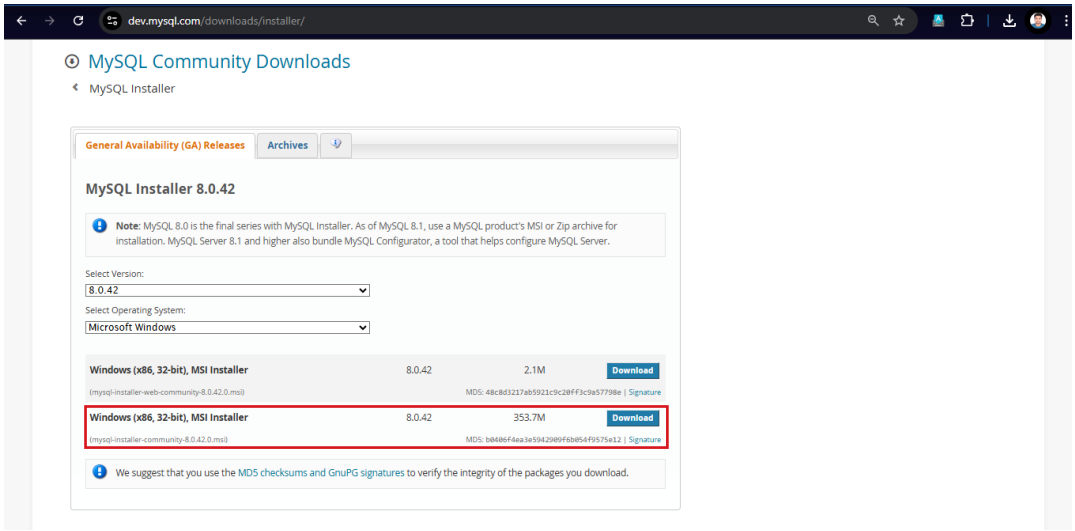


Figure: 2.9

Step 5: Now, Click on **No thanks, just start my download.**

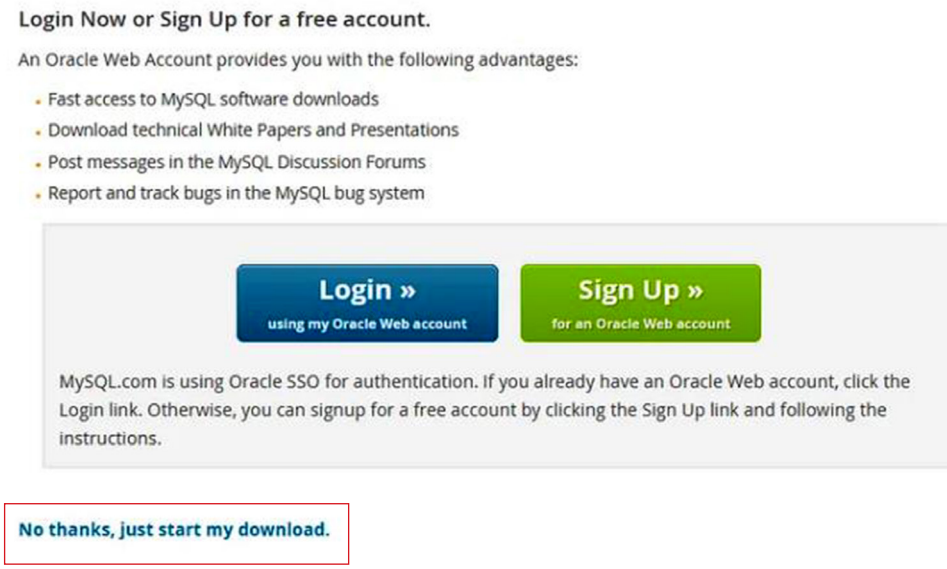


Figure: 2.10

Now, downloading process will proceed. After you download MySQL on your PC, you need to install it into your PC.

Step 6: Locate the installer file you just downloaded (e.g., mysql-installer-web-community-x.x.xx.x.msi) and double-click it to run the installer.

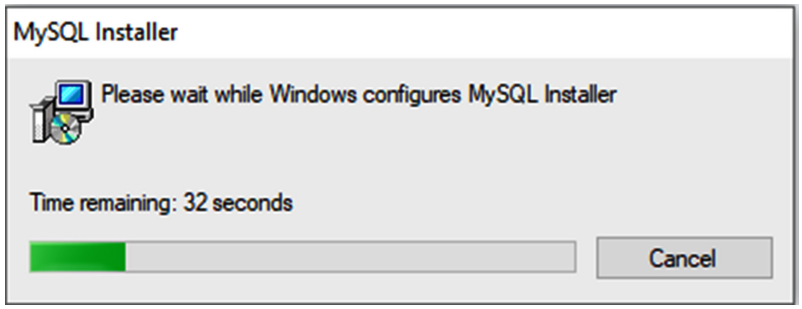


Figure: 2.11

Step 7: In the installer window, choose "Full" under "Setup Type." This option will install MySQL Server, MySQL Workbench and other components, such as documentation and examples. Then click "Next."

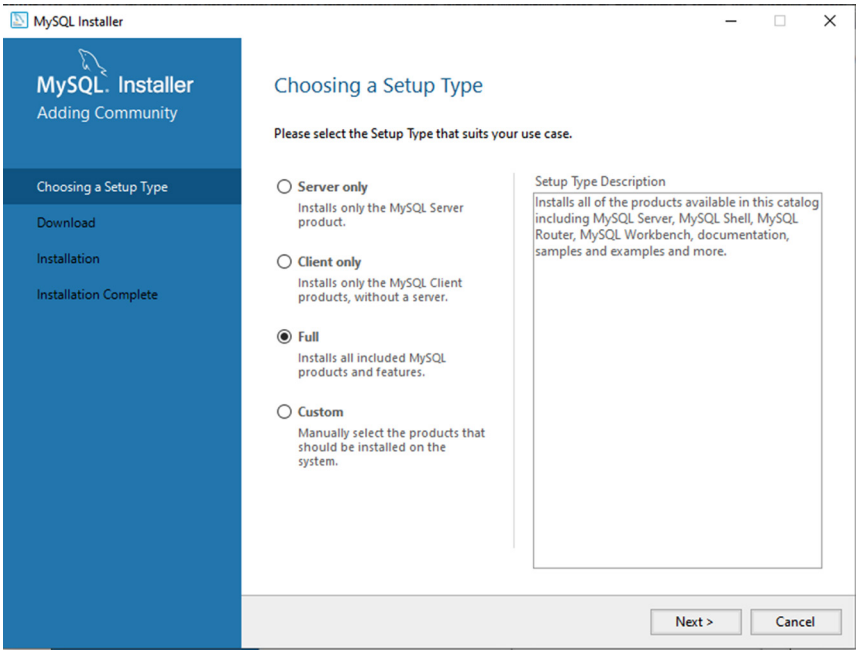


Figure: 2.12

Step 8: In the next window, just click "Execute." After all products have been installed successfully, click "**Next**" to start the configuration.

Step 9: Under "Type and Networking", choose "Development Computer" as Config Type and leave everything else as is. Then, click **"Next."**

Step 10: Authentication method page appears, simply click on **Next**.

Step 11: In the **"Account and Roles"** window, choose a new password for your root account. Be sure to choose a strong password. Remember this password is used when you want to open MySQL using root login.

Step 12: After entering password, MySQL shows **Windows Service** leave everything as is and click **Next.**

Step 13: Next, grant full access to let the Installer update server file permissions and click "Next." Click "Execute" to apply the configuration changes.

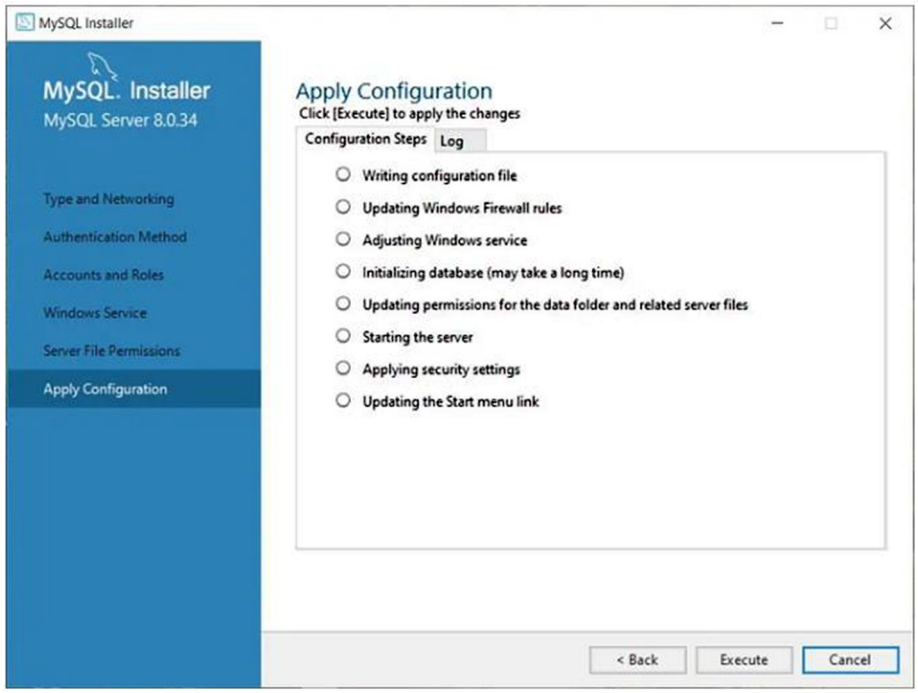


Figure: 2.13

Before finishing, MySQL may ask you to test your connection. Click "Test Connection" to verify that MySQL Server is running properly.

Step 14: Once the configuration is done (green checkmarks), click "Finish."

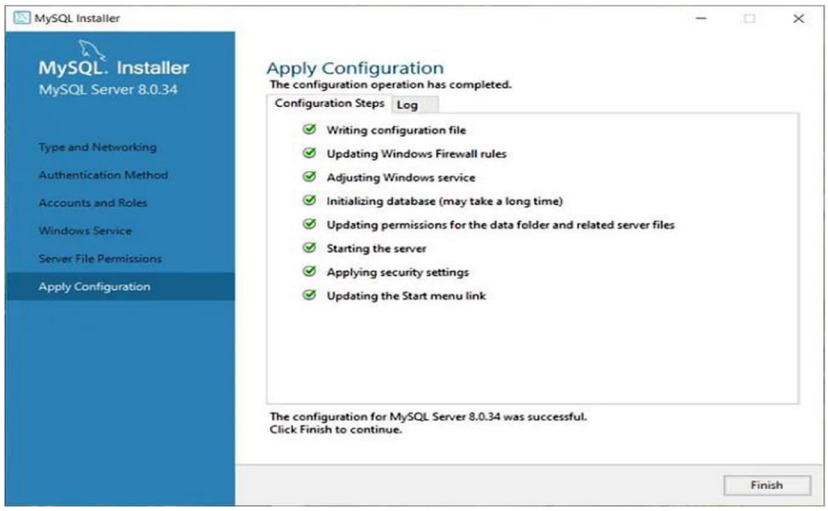


Figure: 2.14

Using MySQL Workbench

At the end of the installation process, MySQL Workbench should launch automatically. If it doesn't, start the program manually. You are greeted by the graphical user interface (GUI) tool for MySQL, the MySQL Workbench.

In the bottom half of the window under "MySQL Connections" you should see one tile for your local instance connection, Local instance MySQL80.

a. Go ahead and double-click it.

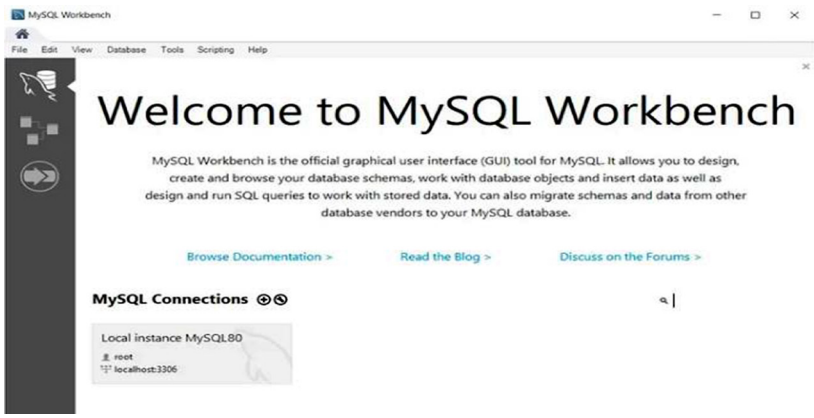


Figure: 2.15

b. In the new window, enter the password you chose earlier and click “OK.”

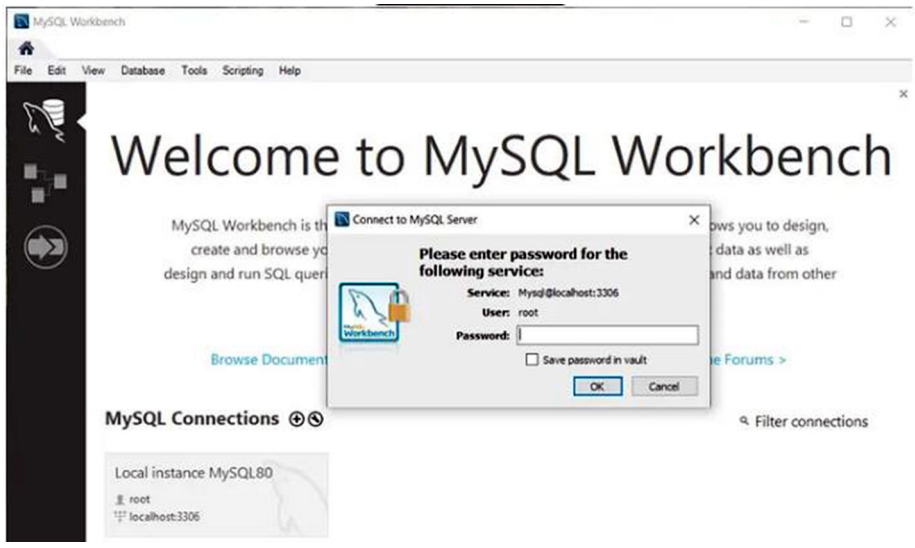


Figure: 2.16

c. You’re now connected to your MySQL Server and ready to start using MySQL Workbench to work with your databases.

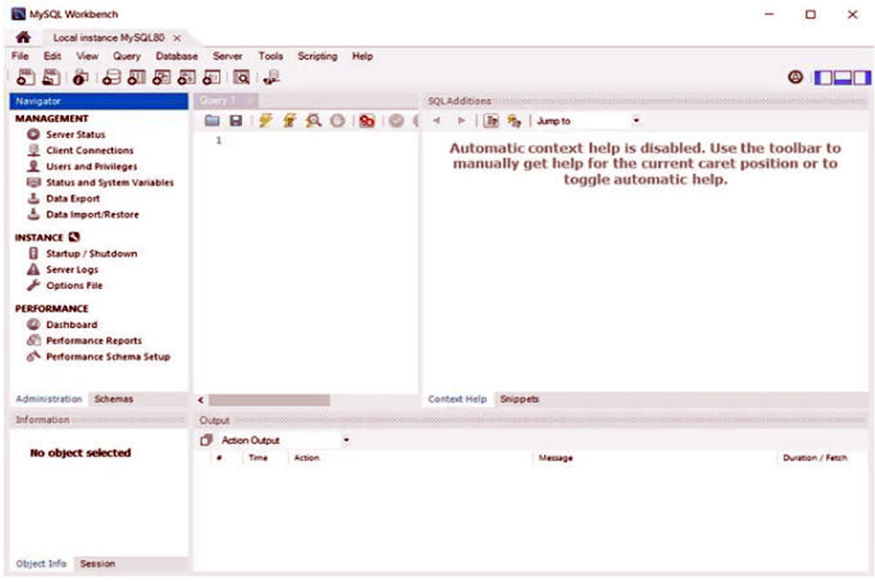


Figure: 2.17 Workbench Interface

Verify MySQL Installation

After installing MySQL, it's important to check whether the installation was successful. Verifying MySQL ensures that the database software is correctly set up and ready to use. To do this, open the command prompt (or terminal) and type:

mysql - - version

If MySQL is installed properly, this command will display the installed version of MySQL. For example: mysql Ver 8.0.34 for Win64 on x86_64

This means MySQL is installed and recognized by the system. You can also try logging in by typing:

mysql -u root -p

This will prompt you to enter the root password. If you successfully log in and see the **mysql > prompt**, it means your MySQL server is running properly. This verification step ensures you are ready to start creating databases and tables.

You can close the MySQL command-line client and return to the system shell prompt by using the **exit;** or **quit;** command, or by pressing **Ctrl + D**.

2.6.1 Table in MySQL

In MySQL, a table is a basic structure used to store data in a database. It organizes information into rows and columns, where each row represents a record and each column represents a field. Tables make it easy to manage and retrieve related data efficiently. All the data in a database is stored in one or more tables. You can create, modify, or delete tables using SQL commands.

Features of a table:

i. Stores data in rows and columns format.

ii. Each column has a specific data type (e.g., number, text, date).

iii. Each row in a table is called a record.

iv. Each column in a table is called a field.

v. Tables can be linked with each other using keys (like Primary Key and Foreign Key).

vi. Tables can be created, modified, or deleted using SQL queries.

Relational Table

A **relational table** is a table in a **relational database** where data is organized into rows and columns, and each table represents one type of entity (like students, teachers, or courses). What makes it *relational* is the ability to **connect (relate)** one table with another using **keys** such as **Primary Key** and **Foreign Key**.

Each row in a relational table is a unique record, and each column represents a specific field of data. Relationships between tables help avoid data duplication and make data handling more efficient and logical.

For an example, you might have:

i. A 'students' table with student details.

ii. A 'courses' table with course details.

iii. A 'student_courses' table that relates students to the courses they are enrolled in.

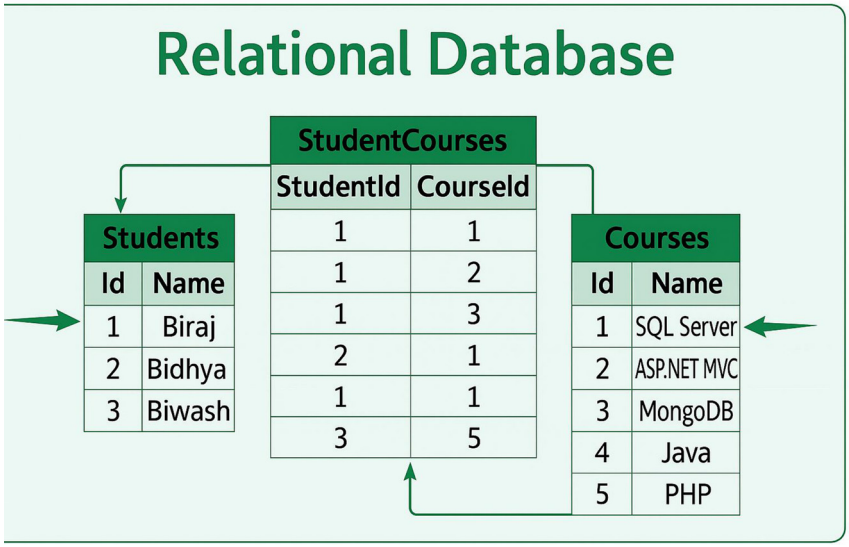


Figure: 2.18

The tables are related using keys, forming a relational database system.

2.6.2 Structured Query Language (SQL)

Simply, a query is a question to the database. SQL commands are the fundamental building blocks for communicating with a database management system (DBMS).

Query is a request made to a database to retrieve, insert, update, or delete data. Queries are written using SQL (Structured Query Language), which was developed by IBM in the 1970s to manage data in relational databases. With the help of queries, users can interact with the database to get meaningful information based on specific conditions.

For example, if you want to find all students in Class 10 from your database, you can write a query to display only those records. Queries allow users to filter, sort, and search large amounts of data quickly and accurately. They also help in updating or removing data without manually searching each record. SQL queries can be simple or complex, depending on the need. Overall, queries are the main way users communicate with a database system to perform useful tasks.

SQL (Structured Query Language) pronounced as **"See"-"quell**" is made of three sub languages DDL, DML and DCL.

Features of a Query

i. Retrieves specific data from tables.

ii. Adds, updates, or deletes records

iii. Filters and sorts data easily

iv. Useful for reports and analysis

MySQL Constraints

MySQL Constraints are rules applied to table columns to control the type of data that can be stored in a database. They help maintain data integrity by ensuring that only valid, consistent, and accurate data is entered into tables. When constraints are used properly, they prevent errors like duplicate values, missing important data, or incorrect entries.

Types of MYSQL constraints and their functions

Constraint	Description
PRIMARY KEY	Ensures no duplicate of NULL values are allowed in the specified column
FOREIGN KEY	Links one table to another using a key from another table, maintaining data consistency.
NOT NULL	Prevents columns from having NULL (empty) values.
UNIQUE	Ensure that all values in a column are different from each other.
CHECK	Limits the values that can be inserted into a column using specific conditions.
DEFAULT	Automatically assigns a predefined value is provided during data insertion.

Importance of Constraints

i. Maintain data accuracy and consistency

ii. Prevents duplicate or incorrect data entries

iii. Establish relationships between tables

iv. Enforce business rules and conditions within the database.

2.7 Data Definition Language (DDL)

DDL (Data Definition Language) is a part of SQL that is used to define and manage the structure of database objects like tables, schemas, indexes, and views. It helps in creating, modifying, and deleting the physical structure of the database. DDL statements do not handle data directly, they deal with how the data is stored and organized and used to define the physical characteristics of records.

When a DDL command is executed, it automatically saves (commits) the changes to the database, meaning those changes cannot be rolled back.

Here are the common DDL commands and their functions.

Commands	Description
CREATE	Creates new database objects such as tables, views, or databases.
ALTER	Changes the structure of an existing table (e.g., add a column).
DROP	Deletes an existing database object like a table or database permanently.
TRUNCATE	Removes all records from a table but keeps the table structure.
RENAME	Changes the name of an existing table or column.

1. CREATE Query

The **CREATE** query is part of DDL (Data Definition Language) in SQL. It is used to create new database objects such as databases, tables, views, and indexes.

Before creating tables, we must first create a database where the tables will be stored. SQL provides the CREATE DATABASE command to make a new database.

The CREATE DATABASE Statement is used to create a new database. After creating a database we should select it by using USE command.

Syntax:

CREATE DATABASE database_name; Example: Create database school;

Where, database_name: The name you want to give to your new database.

The CREATE TABLE Statement is used to create a new table in a database.

Syntax

```
CREATE TABLE table_name (
  column1   datatype,
  column2  datatype,
  column3  datatype,
  ...
);
```

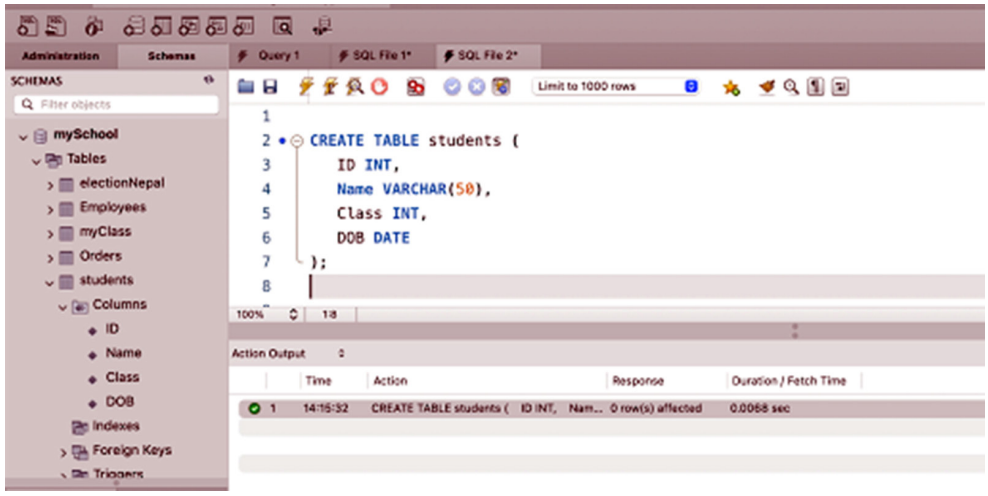
Where, column1,column2,column3 …. are columns of the tables and data types are the types of data that is held by a specific column.

Activity 1 : Create Table

```
CREATE TABLE students (
  ID INT,
  Name VARCHAR(50),
  Class INT,
  DOB DATE
);
```

Explanation: This command creates a table named students with four fields.

Demonstration:



Activity 2 : Adding Primary Key while creating table

```
CREATE TABLE students (
  ID   INT PRIMARY KEY,
  Name  VARCHAR(50),
  Class  INT,
  DOB    DATE
);
```

Explanation: The PRIMARY KEY ensures that each student has a unique ID and that the ID field cannot be left empty (NULL).

Activity 3 :

Adding Not Null while creating Table

```
CREATE TABLE students (
  ID  INT PRIMARY KEY,
  Name  VARCHAR(50),
  Class  INT NOT NULL,
  DOB  DATE
);
```

Explanation: Class column is marked as NOT NULL, meaning it must have a value when inserting a record.

2. ALTER Query

The ALTER query in SQL is used to modify the structure of an existing table. It allows you to:

a. To add new column

Syntax :

```
ALTER TABLE  table_name
ADD  column_name  data_type;
```

b. To delete/drop a column

Syntax:

```
ALTER TABLE  table_name
DROP COLUMN  column_name;
```

c. To modify of a column

Syntax :

```
ALTER TABLE table_name
MODIFY COLUMN  column_name  new_data_type;
```

d. Rename a column

Syntax :

```
ALTER TABLE table_name
CHANGE old_column_name   new_column_name   data_type;
```

e. To remove constraints

Syntax :

```
ALTER TABLE  table_name
DROP CONSTRAINT  constraint_name;
```

3. DROP Query

The DROP query in SQL is used to permanently delete a database object such as a table, database, or view from the database system. Once dropped, the object and all its data cannot be recovered. Use the DROP command carefully!

Syntax: To drop a Database

DROP DATABASE database_name;

Syntax: To drop a table

DROP TABLE table_name;

2.8 Data Manipulation Language (DML)

DML (Data Manipulation Language) is a part of SQL that is used to work with the data stored in database tables. It helps users insert new records, retrieve existing data, update values, and delete unwanted records from the database. Unlike DDL, which deals with the structure, DML focuses directly on the content of the database.

DML statements allow users to manage and manipulate data efficiently without changing the structure of the table. These operations can be rolled back if needed, which means changes are not saved permanently unless a COMMIT command is used.

Here are the common DML commands and their functions.

Command	Description
INSERT	Adds new records into a table
SELECT	Retrieves data from one or more tables
UPDATE	Modifies existing data in a table
DELETE	Removes records from a table

1. INSERT Query

An INSERT query in SQL is used to add new records (rows) into a database table. Think of it as the way you put fresh information into your database. You can use it to add a single row of data or multiple rows in one go. When you execute an INSERT query, you're essentially providing the values for each column in the new row(s).

Syntax

INSERT INTO table_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

INSERT INTO table_name

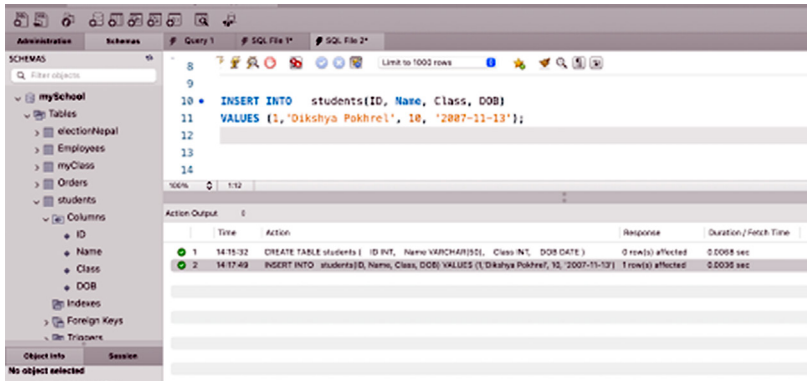
VALUES (*value1*, *value2*, *value3*, ...);

Activity 5 :

a) INSERT INTO students(ID, Name, Class, DOB)
VALUES (1, 'Dikshya Pokhrel', '10', '2007 ');

This INSERT query adds a new row into the students table. It specifies the values for the Id, Name, Class, and DOB columns as 1,'Dikshya Pokhrel', '10', and 1990 respectively.

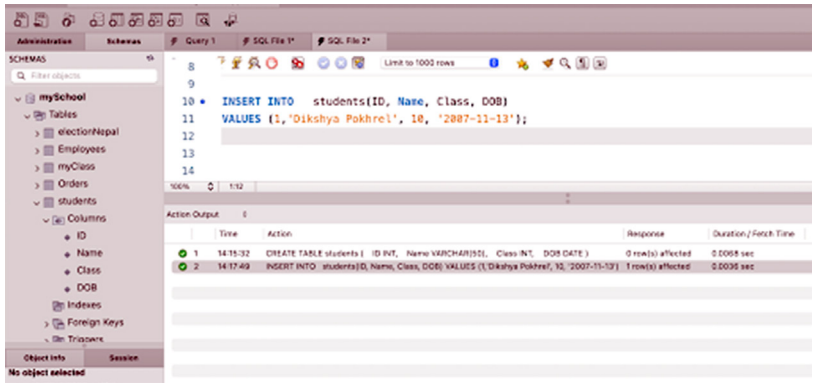
Demonstration:



b) INSERT INTO students(ID,Name) VALUES (2, 'Binay Adhikari');

This query adds a new row to the students table. Since column names are not specified, 2 and 'Binay Adhikari' are inserted into the specified table's columns.

Demonstration:



2. SELECT Query

A SELECT query in SQL is used to retrieve data from a database table. It helps you view specific information you need, such as all rows, selected columns, or rows matching certain conditions. The data returned is stored in a result table, called the result-set.

Syntax

SELECT column(s)*_name*

FROM table_name;

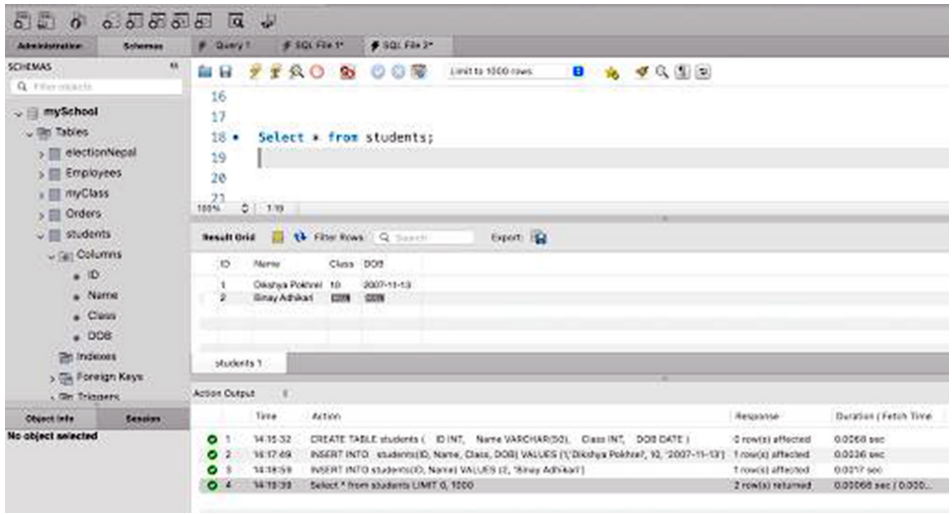
Activity 4 :

a) *SELECT* * *FROM* students;

This query display all records from a table students;

We can use the * character to select ALL columns from a table.

Demo:



b) *SELECT* Name *FROM* students ;

This query displays all Names inserted in a students table.

c) *SELECT* name, DOB *FROM* record;

This query displays all name and DOB column's data from a students table.

3. UPDATE Query

An UPDATE query in SQL is used to modify existing records (rows) within a database table. Think of it as a way to change or correct information that's already in your database. You can update specific columns in one or more rows based on certain conditions.

Syntax:

UPDATE table_name

SET column1 = value1, column2 = value2, …

WHERE condition;

Activity 6:

a) *UPDATE* student_tbl
SET grade = 'B'
WHERE name = 'JaiShree';

This query updates the grade of the student named 'Jaishree' in the student_tbl to 'B'.

b) UPDATE account_tbl
SET balance = balance + 100, last_transaction = CURRENT_TIMESTAMP
WHERE account_number = 1001;

This query increases the balance of the account with account_number 1001 by 100 and updates the last_transaction column to the current date and time.

c) *UPDATE* record
SET rank = rank + 1
WHERE rank <= 3;

This query increases the rank by 1 for all records in the record table where the current rank is 3 or less.

4. DELETE Query

A DELETE query in SQL is used to remove existing records (rows) from a database table. This functionality enables the removal of information that is deemed unnecessary or outdated. You can delete specific rows based on a condition or remove all rows from a table.

Syntax

DELETE FROM table_name

WHERE condition;

Activity 7 :

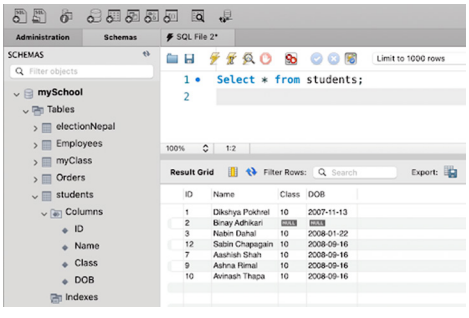
a) DELETE FROM students
WHERE name = 'Muna';

This query deletes the row from the students where the name is 'Muna'.

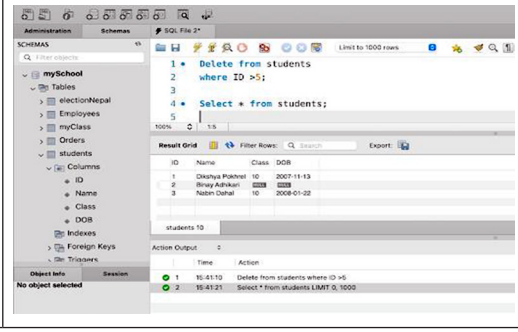
b) DELETE FROM students
WHERE ID > 5;

This query deletes all rows from the student table where the rank is greater than 5.

Before using Delete query



After Using Delete Query



c) DELETE FROM students;

This query will remove all rows from the students. The table structure (columns, data types, constraints) will remain, but it will contain no data.

WHERE and LIKE Clauses

Using the SELECT query, we can retrieve or extract data from the database, based on specific criteria by using the WHERE and LIKE clauses.

a. Where Clauses

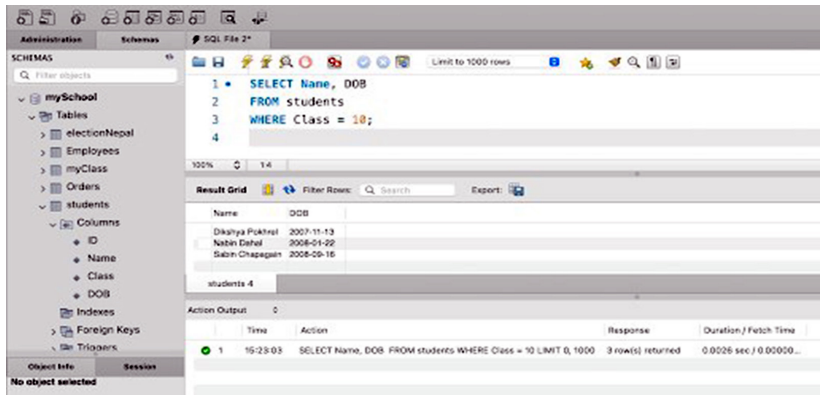
The WHERE clause in MySQL is used to **filter records** and retrieve only those rows that meet specific conditions. It helps in selecting data based on a given rule rather than displaying all records from a table. Without a WHERE clause, SQL queries would return the entire table.

Activity 8 :

a)
```
SELECT Name, DOB
FROM students
WHERE Class = 10;
```

This query retrieves the Name and Date of Birth of students who are in Class 10. It filters the results to show only students from that specific class.

Demonstration:



b)
```
UPDATE students
SET Class = 11
WHERE ID = 5;
```

This query updates the Class of the student whose ID is 5, changing it to 11. Only the record with ID 5 will be modified, and other records remain unchanged.

c)
```
DELETE FROM students
WHERE Name = 'Biraj';
```

This query deletes the record where the Name is exactly 'Biraj'.

b. Like Clauses

The LIKE clause is used in SQL to search for a specified pattern in a column. It is often used with the WHERE clause to filter records based on partial matches instead of exact values.

LIKE works with wildcard characters:

% (percent sign) : Represents zero, one, or multiple characters.

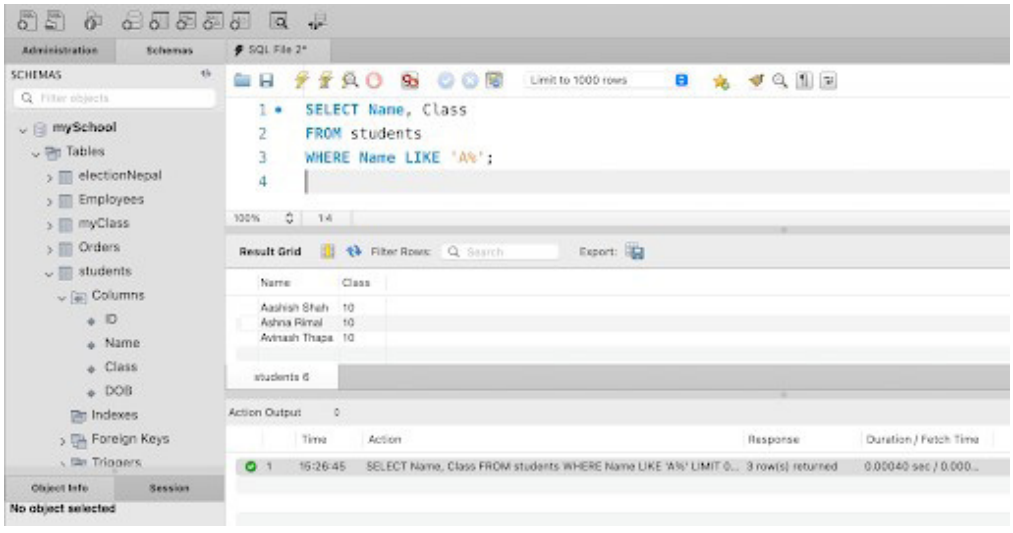
_ (underscore): Represents a single character.

Wildcard characters in MySQL are special symbols used with the LIKE clause to represent unknown parts of a string. They help in searching for data when you don't know the exact value but know a pattern.

Activity 9 :

a) SELECT Name, Class
FROM students
WHERE Name LIKE 'A%';
This query selects the Name and Class of all students whose names start with the letter "A". The % wildcard means "any number of characters" after A.

Demonstration:



b) UPDATE students
SET Class = 12
WHERE Name LIKE '_man';
This query updates the Class to 12 for students whose names have exactly four letters and end with "man", like "Aman" or "Iman".
The _ wildcard represents a single unknown character before "man".

c) DELETE FROM students
WHERE Name LIKE '%esh';
This query deletes all student records where the Name ends with "esh", such as "Rajesh" or "Ramesh".
The % wildcard means "any characters" before "esh".

2.8.5 Report

In MySQL, a report refers to the organized presentation of data retrieved from one or more tables using SQL queries. Although MySQL itself does not directly create designed, graphical reports (like charts or printed layouts), it provides the raw structured data which can be used to generate reports.

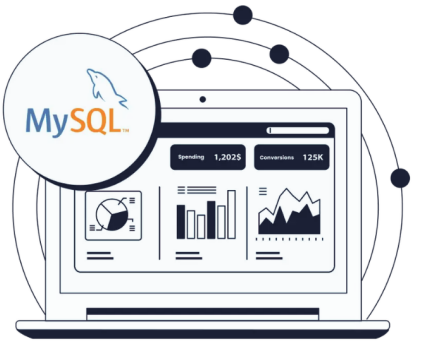


Figure: 2.19

Reports help users to understand, analyze, plan and make decisions based on the collected data.

In simple terms, a report in MySQL is the result of a query that extracts specific information from the database and presents it in a readable and meaningful format.

By using SQL queries, we can create basic reports directly from MySQL.

For more advanced graphical or printable reports, MySQL can be connected to reporting tools like Power BI, Tableau, or Google Data Studio.

Types of Reports You Can Generate with MySQL Queries:

i. **Summary reports:** Show total values like total sales, number of students, etc.

ii. **Detail reports:** Display detailed records (e.g., list of all students with marks).

iii. **Grouped reports:** Group data by categories like department, class, city, etc.

iv. **Filtered reports:** Reports based on specific conditions (e.g., students above 80% marks).

v. **Aggregate reports**: Use functions like SUM, AVG, COUNT to summarize data.

Features of Reports in MySQL

i. Organized data presentation

ii. Data Filtering

iii. Summarization

iv. Multi-Table Combination

v. Timely Access

vi. Easy Export

Case Study

You have been hired as a database designer for a mobile farming application named **KrishiKranti**, developed to help Nepali farmers manage their crops, tools, sales, and farm activities efficiently.

Your task is to design a relational database that supports the core functions of the app based on the requirements and real-world farming scenarios. You may refer to doanepal.gov.np/ne/ for agricultural data.

Your case study must contain following criteria:

a. Introduction

b. Problem Statement

c. Objective

d. SQL Integration

e. Expected Outcomes

f. Conclusion

Students need to submit a hard copy of case study file to the teacher.

Project Work

Project Title : Database Design for "KrishiKranti" Farming App

Design a database for a farming app named KrishiKranti that helps farmers manage crops, tools, sales, and farm activities.

Your task:

a. Planning: Identify at least 5 entities related to the app (e.g., Farmer, Crop, Tools, Sales, FarmActivity).

b. Database Design in MySQL:

   Create each table using proper data types and constraints (e.g., NOT NULL, UNIQUE, DEFAULT, CHECK, etc.)

   Use foreign keys to show relationships between entities.

   Insert at least 25 records in each table.

   Ensure proper normalization and meaningful field names.

Note: You can take reference from **https://doanepal.gov.np/ne/** which is the URL of 'Ministry of Agriculture and Livestock Development for Farming' for related information that can be useful to design your database.

Exercise

1. Choose the correct answer.

a. Which of the following is NOT a function of a DBMS?

i. Data storage
ii. Data retrieval
iii. Word processing
iv. Data manipulation

b. Which type of key uniquely identifies each record in a table?

i. Foreign Key ii. Composite Key

iii. Primary Key iv. Candidate Key

c. What is the primary purpose of a foreign key?

i. To store only text data.

ii. To uniquely identify each record in its own table.

iii. To establish and enforce relationships between tables.

iv. To combine multiple fields into one.

d. In a one-to-many relationship between teachers and classes, which statement is true?

i. Each teacher teaches only one class, and each class has only one teacher.

ii. Each teacher can teach multiple classes, and each class has only one teacher.

iii. Each teacher teaches only one class, and each class can have multiple teachers.

iv. Each teacher can teach multiple classes, and each class can have multiple teachers.

e. Which SQL command is used to add new rows to a table?

i. SELECT ii. UPDATE

iii. INSERT iv. DELETE

f. Which SQL command is used to modify existing data in a table?

i. SELECT ii. UPDATE

iii. INSERT iv. DELETE

g. Which SQL clause is used to specify conditions for data retrieval?

i. SELECT ii. FROM

iii. WHERE iv. ORDER BY

h. What does the SQL LIKE clause help you do?

i. Perform extract matches.

ii. Search for patterns in data.

iii. Sort the data.

iv. Define data types.

i. Which SQL data type is used for storing only date values (e.g., YYYY-MM-DD)?

i. TIME ii. DATETIME

iii. TIMESTAMP iv. DATE

j. Which SQL data type is suitable for storing names and addresses, allowing for variable length?

i. CHAR ii. TEXT

iii. INT iv. VARCHAR

k. What is the primary difference between CHAR(n) and VARCHAR(n) data types?

i. CHAR stores only numeric data, while VARCHAR stores text.

ii. CHAR has a fixed length, while VARCHAR has a variable length up to n characters.

iii. VARCHAR is used for dates, while CHAR is for text.

iv. There is no significant difference between them.

2. Write short answers to these questions.

a. Define data and information.

b. What is primary key?

c. What is the foreign key?

d. Define fields and rows.

e. Differentiate database and DBMS.

f. Describe RDBMS.

g. Why is a DBMS considered more advantageous for managing large amounts of data compared to a simple spreadsheet?

h. Describe the purpose of primary and foreign keys in a relational database.

i. Explain the concept of a one-to-many relationship and provide a real-world example (different from the one in the text).

j. Briefly describe the roles of DDL and DML in SQL.

k. Explain the function of the WHERE clause in a SELECT statement.

l. What are SQL constraints? Why are they important for maintaining data integrity?

m. What is a composite key, and when might it be necessary to use one?

3. Write long answers to these questions.

a. Define DBMS with its advantages.

b. What is the importance of primary key in db? List out its features.

c. Differentiate between DDL and DML.

d. Show difference between Table and Query.

e. What is the relationship in DBMS? Explain its types.

f. What is a Query? List out its importance in DBMS.

g. What is a report in the context of MySQL and what is its purpose?

h. Describe report with its features.

4. Write down the SQL query for the following statement.

a. Write an SQL query to create a new database named CompanyData.

b. Write an SQL query to create a table named Employees with the following columns:

- EmployeeID as an integer (INT) and the primary key.
- FirstName as a variable-length string (VARCHAR) with a maximum length of 50 characters.
- LastName as a variable-length string (VARCHAR) with a maximum length of 50 characters.

- Salary as a decimal number (DECIMAL) with a total of 10 digits and 2 digits after the decimal point.

c. Write an SQL query to add a new column named Email to the Employees table. The data type for this column should be a variable-length string (VARCHAR) with a maximum length of 100 characters.

d. Write an SQL query to modify the Salary column in the Employees table to have a default value of 25000.

e. Write an SQL query to rename the Employees table to StaffMembers.

f. Write an SQL query to insert a new record into the StaffMembers table with the following values: [EmployeeID : 101, FirstName : Sampada, lastName : Bhattrai, Salary : 30000.50]

g. Write an SQL query to retrieve all columns and all rows from the table.

h. Write an SQL query to retrieve only the FirstName and Salary columns from the Employees table.

i. Write an SQL query to increase the salary of all employees by 10%.

j. Write an SQL query to delete the employee with EmployeeID 103 from the table.

k. Write an SQL query to delete all records from the Employees table (or StaffMembers), but keep the table structure intact.

Unit 3 Multimedia




Let's think about it:

1. *How can multimedia help make communication more engaging?*
2. *Image learning new content without the use of images, text or videos?*
3. *What makes multimedia elements in movies, advertisements, and games appealing?*

Introduction

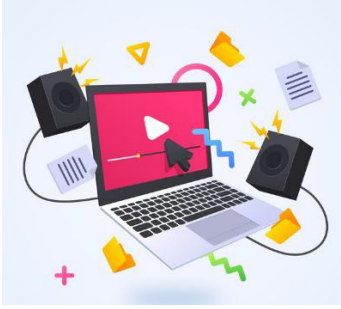


Figure 3.1

Imagine you're using an educational app on your mobile phone to study. The app shows animated science experiments, has voice explanations, practice quizzes, and even video lectures by teachers from all over Nepal. All these different types of media such as text, images, audio, video, and animation, are integrated into one

platform. This is a real-life example of multimedia, and the tools and technologies used to create that app fall under multimedia technology.

There are many other apps that uses multimedia technology. Some of them are:

Duolingo - An app that uses voice, text, sound effects, and animation to teach different languages

TikTok - Popular for its content that mixes video, filters, text, and music to create short videos

YouTube - Online streaming platform that combines video, sound, thumbnails, and live chat for learning and entertainment

Google Earth - It combines satellite images, text labels, and even 3D maps for exploration.

Do you know?

More than **80% of online content traffic today is video**, which is one of the richest multimedia formats. (source https://www.demandsage.com/video-marketing-statistics/)

The term multimedia is made up of 2 words "**multi**" and "**media**". "**Multi**" refers to many while "**media**" refers to different types of communication forms. Hence, the term "**multimedia**" refers to different types of media.

The technology that uses these different types of media in order to create multimedia content like movies, games, and websites is known as multimedia technology.

The different components of multimedia are as follows:

a. Text

b. Graphics

c. Audio

d. Video

e. Animation

Applications of Multimedia

There are different uses or applications of multimedia and some of the major ones are in the field of education, entertainment, business and communication.

a. **Education -** Learning contents such as e-books, online courses and simulations utilise multimedia as they can create realistic and interactive experiences among learners for better understanding.

b. **Entertainment -** Video games, movies, music and virtual reality, all use multimedia. Due to this we have interactive experiences all around us and have changed the way of entertainment.

c. **Business -** Advertisements that we see all around us are all the result of multimedia that helps to capture audiences' attentions. This in turn results in effective marketing for businesses.

d. **Communication -** Video conferencing and social media that we use for communication use multimedia contents. This in turn make it effective for communication among our friends, family and colleagues.

e. **Content creation -** Social media platforms heavily rely on multimedia tools to create engaging and visually appealing content. Multimedia elements such as videos, graphics, animations, and sound effects help creators deliver messages more effectively and creatively, attracting larger audiences and boosting online presence.

Do you know?

The term "multimedia" was first used in 1966 to describe a performance combining art, music, and technology.

3.1 Text

Text refers to the **characters** that we see on screen. It is also referred to as **alpha numeric** values. Combination of such characters gives meaningful information to the readers.

There are many types of text editors and some common ones that we use are Notepad, WordPad and Microsoft Word.

Text is a part of multimedia as it can be presented in various ways. By using multimedia software, text can be presented in different sizes, style or colour.

3.1.1 Text File Formats

a. **TXT** - TXT is a standard Text file format that only stores text contents.

b. **RTF** - It stands for Rich Text Format and it can store text content along with text styles like colour, line spacing, size, indentation and so forth.

c. **DOCX** - It is the native document format of Microsoft Word and can store text along with text styles just like RTF file format.

Do you know?

DOCX, introduced by Microsoft in 2007, is now the most widely used word processing format!

3.2 Graphics

Graphics refers to the **visual elements** that we see on a screen. Graphics may include different shapes in different colours through the combination of which we get information visually. It also refers to the **images** or **pictures** that we see on screen.

There are many types of graphics editors and some common ones that we use are Pixlr X, Adobe Photoshop and Adobe Illustrator.

By using such applications, we can edit various shapes, add effects to them and edit images as we like.

3.2.1 Raster and Vector Graphics

Graphics are stored in a standard digital format. Computer graphics can be categorised by 2 types, **Raster Graphics** and **Vector Graphics**.

Raster Graphics

The Raster format of graphics is also known as **Bitmap** Graphics. These kinds of computer graphics are made up of pixels.

A pixel stands for Picture Element and it is the smallest unit or elements of a picture that is displayed on a screen.

Pixels refers to the collection of tiny square dots that make up an image. Here each pixel has its own colour through the collection of which detailed pictures or graphics are formed. Therefore the more number of pixels there are on an image, the higher level of details it has.

Common Raster Graphics File Formats

There are many raster graphics file formats that are used, but the most common ones among them are PNG, GIF, and JPG.

PNG - PNG stands for Portable Network Graphics. It is one of the commonly used formats as it supports transparency for images and graphics.

GIF - GIF stands for Graphics Interchange Format. It is the only image format that supports animations and that is why it is one of the commonly used formats.

JPG - JPG also known as JPEG stands for Joint Photographic Experts Group. It is the most commonly used format, as it can store high quality images with very good compression, which results in a low file size.

Vector Graphics

Vector graphics format is a type of computer graphics that are made up of Bezier curves. These curves are stored in the form of mathematical formulas.

Due to this very reason, the quality of the vector graphics remains the same no matter what size it is in.

Common Vector Graphics File Formats

There are various vector graphics file formats, and the most commonly used one is SVG.

SVG - SVG stands for Scalable Vector Graphics and is a graphics format that is commonly supported by most applications.

PDF File format

PDF Stands for Portable Document Format and it is one of the versatile graphics formats. PDF can contain both raster and vector graphics depending on how it was saved and by which application it was saved.

This format is commonly used to share documents online, but it can also store images and graphics.

3.2.3 Graphics Editing

In order to start editing Graphics, you can use Pixlr X. Pixlr X is a free online tool that lets you edit photos and graphics easily. You can also download and use an app version of it, named Pixlr AI Photo Editor, on your mobile phone. You can crop, resize, rotate, adjust colors, add filters, and more. Pixlr X is a great way to start learning about graphics editing.

Before starting to work in Pixlr X, you need to know that there are four main areas of the application. They are, the Canvas area, tools panel, layers panel and navigational tools that.you will be using in order to create and edit projects.

Do you know?

GIF is the only image format that supports animations! That's why it's commonly used for short looping animations on websites and social media.

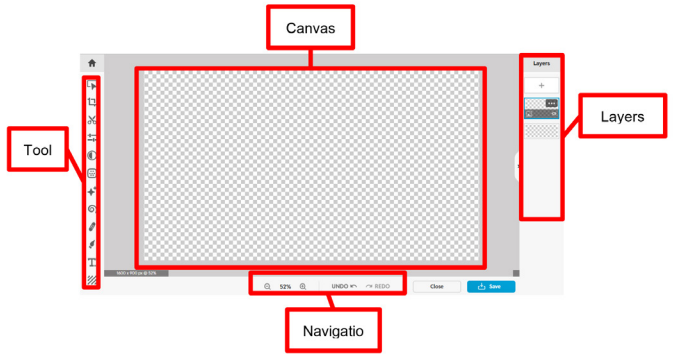


Figure 3.2 Pixlr X Interface

Canvas area: Workspace for viewing, editing images and graphics.

Tools panel: Offers various editing tools, functionalities.

Layers panel: Manages and organizes image, graphics and text layers.

Navigation tools: Area to access zoom, pan, rotate functions.

Creating a New Project

To start using **Pixlr X** first of all, you must create a **new project**.

For that you need to follow the following steps:

Step 1: Go to the **Pixlr** website (**www.pixlr.com**) and click on the **Pixlr Express or Pixlr Editor** button.

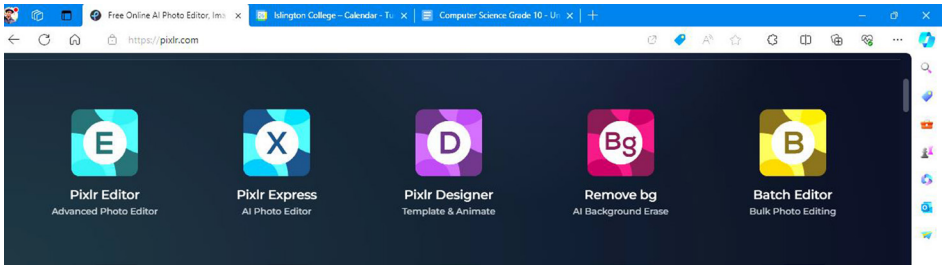


Figure 3.3 Pixlr Homepage

Step 2: New page will open now where you have the option to **open existing image** or **create a new file**.

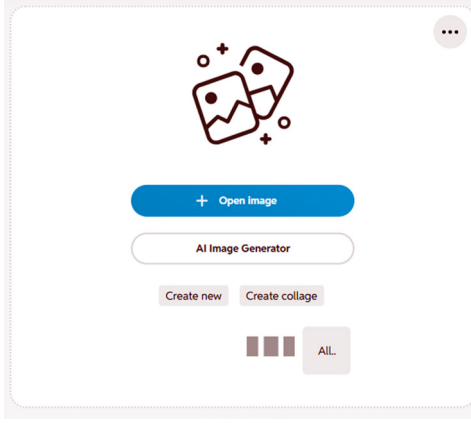


Figure 3.4 New Project Page

Step 3: Choose the option that suits your needs. For example, if you want to start with a blank canvas, choose **Create new.** Then a dialogue box will appear.

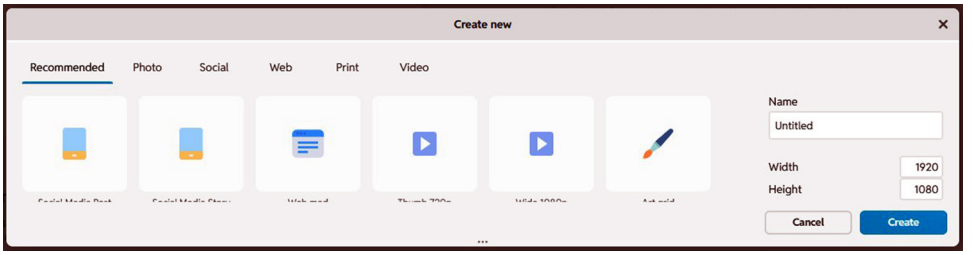


Figure 3.5 Create New Dialogue Box

Step 4: You may choose canvas size from a set of pre-existing **presets**, or you can enter your own **width and height** as well. For example, you can type in a name as **Class 10 Project** in the name section and then enter the width as 1920 by 1080.

Step 5: Then you need to click on the **create button** on the bottom right-hand corner.

Step 6: After that, your project will be created and you may start **creating or editing** contents.

Figure 3.6 Pixlr X Interface

Working with Text

To start editing in Pixlr X, you can start by adding in text. To add and work with text, you can follow these steps:

Step 1: Once a new project is created in Pixlr X, you can start **adding** and **editing text.**

Step 2: To add text, click on the "**Text**" tool on the left side of the screen. It looks like a capital letter "**T**".

Figure 3.7 Text Tool

Step 3: Now click on "**Add new text**". Then, a text box with sample text appears in the **canvas area**.

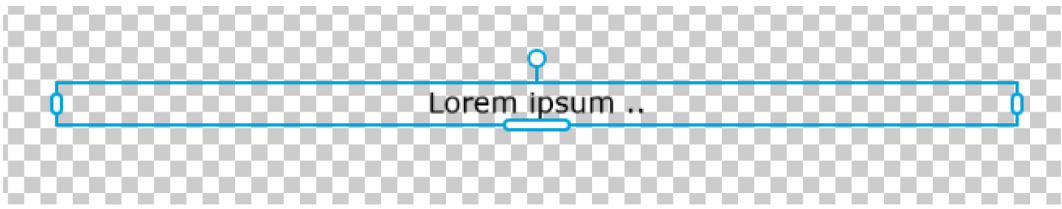


Figure 3.8 Text Box

Step 4: Type your words in the text box. You can also copy and paste text from another source.

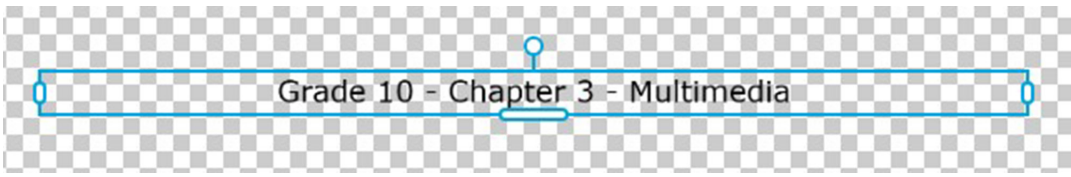


Figure 3.9 Text Box

Step 5: To access font editing options, click on the **arrange button** on the left-hand side when the **text** is selected.

Figure 3.10 Arrange Tool

Step 6: To change the **font, font size, style, colour, alignment, or spacing** of your text, use the options on the left side of the screen.

Figure 3.11 Text Options

Step 7: To resize your text, use the handles on the edges of the text box. You can also use the **arrow keys** on your keyboard when the text is selected or **click and drag** to text to move the text.

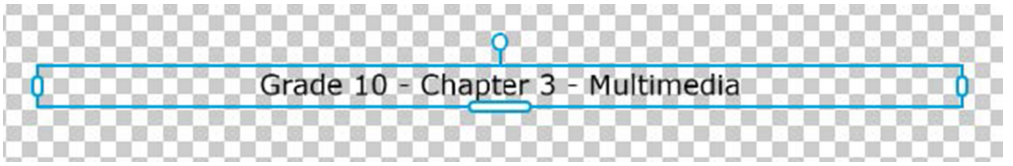


Figure 3.12 Text Box

Step 8: To rotate the text, you can click and drag the **circular dot** on the top of the text.

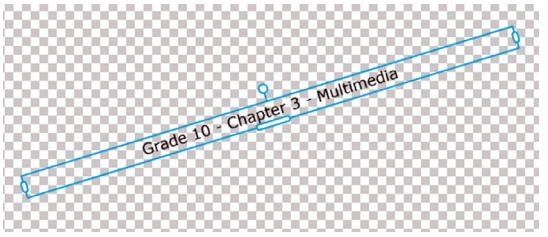


Figure 3.13 Text Box

Step 9: To edit your text later, click on the text layer on the right of the screen. You can also **hide, lock, duplicate, or delete** your text from there.

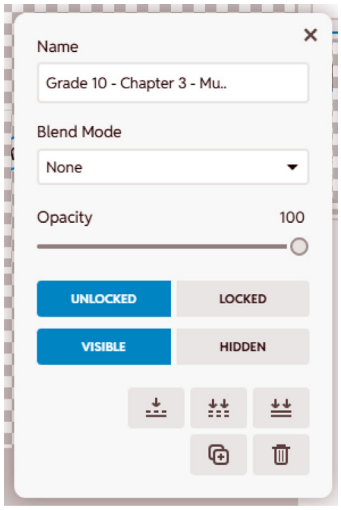


Figure 3.14 Layer Options

Working with Shapes

Another element that you can work with inside of Pixlr X are shapes. So, to add and edit shapes in Pixlr X, you can follow the following steps:

Step 1: Click on the "**Arrange**" tool on the left sidebar.

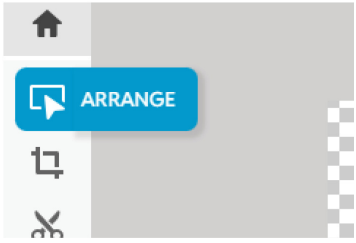


Figure 3.15 Arrange Tool

Step 2: Now, you can see three options that you need to click on "**Shape**".

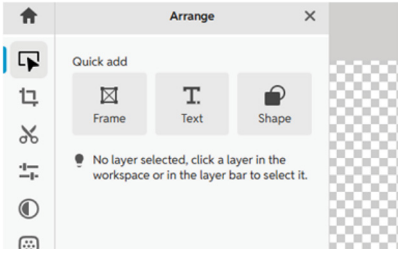


Figure 3.16 Arrange Options

Step 3: Now, a shape will be inserted inside of the canvas area and by default it will be a square shape.

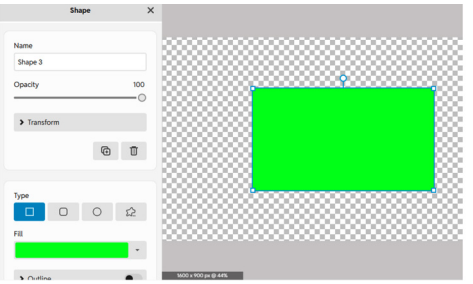


Figure 3.17 Shape

Step 4: You can choose the shape you want to change it to from the **type section**. For example, you can choose "**Circle**" as a shape.

Figure 3.18 Shape Option

Step 5: To edit the shape, you can use the options on the left sidebar. You can change the **fill colour, outline colour** and **shadow options** of the shape.

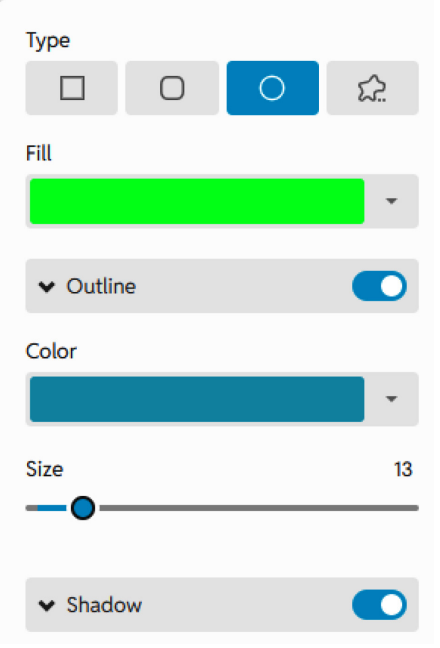


Figure 3.19 Shape Option

Step 6: You can adjust the **size** and **rotation** by dragging the **small square shaped dots** on the **corners** of the shape and using the **circular dot** in the top to **rotate** the **shape.**

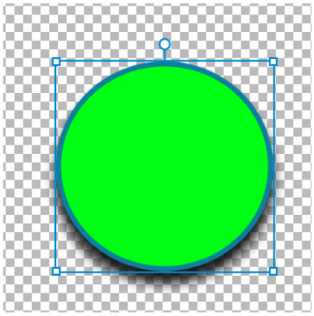


Figure 3.20 Shape

Image Placement - Inserting and Editing

Working with Layers

To manage different elements in Pixlr X, Layers can be utilized. The steps to follow for utilizing layers are as follows.

Step 1: You can access the **"Layers"** section on the right-hand side of the application.

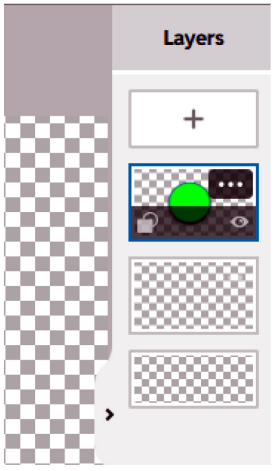


Figure 3.21 Layers Panel

Step 2: Create a new layer by clicking the **"+"** icon.

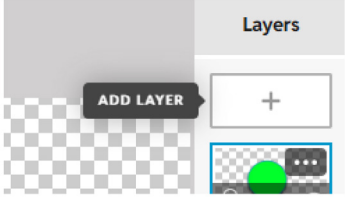


Figure 3.22 Layers Panel

Step 3: After that, "**Add Layer**" dialogue box pops up where you can select different types of layers that you want to add.

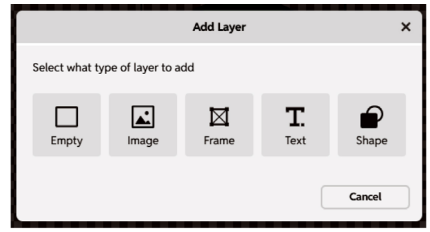


Figure 3.23 Add Layer Dialogue Box

Step 4: You need to click on the **type of layer** that you want to add. For example, you can select a shape **layer**.

Step 5: You can click on the **3-dot** button on a particular layer that you want to work with and a **layer dialogue box** opens.

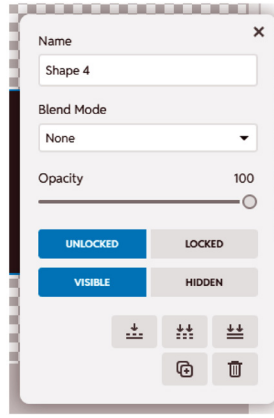


Figure 3.24 Layer Option

Step 6: In the layer dialogue box, you can add a name as you like, Lock or Hide Layer as well.

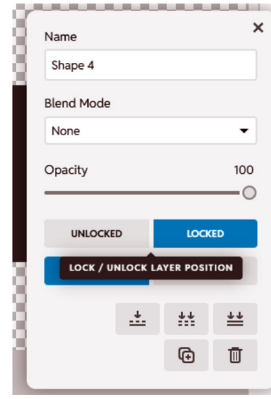


Figure 3.25 Layer Option

Step 7: You can also adjust the transparency and blend mode of each layer by clicking the three dots on the layer.

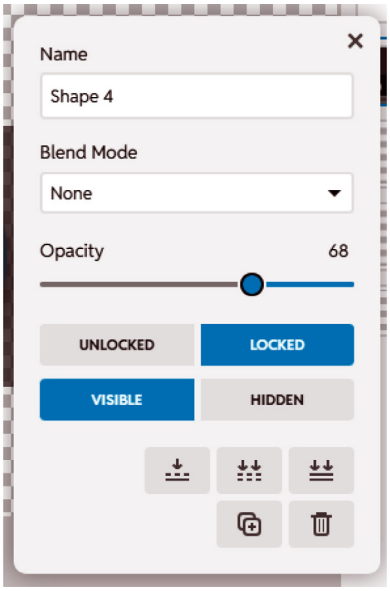


Figure 3.26 Layer Option

Step 8: To rearrange layers, you can click and drag the specific layers you like from the layer's panel.

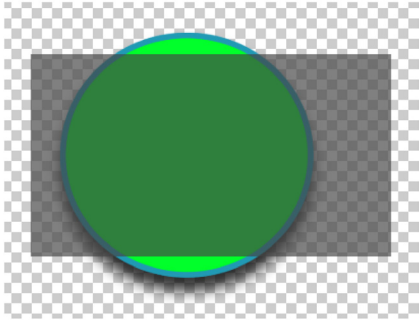


Figure 3.27 Shapes

Inserting Image in Existing Project

You can also insert images in the existing Pixlr X Project. To insert an image in existing project, you have to follow the following steps:

Step 1: Click on the element **icon** in the bottom left hand corner.

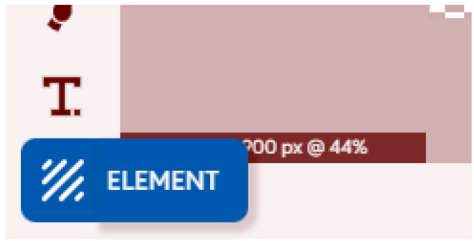


Figure 3.28 Element Tool

Step 2: Then, click on **Add Media** that appears in the element options.

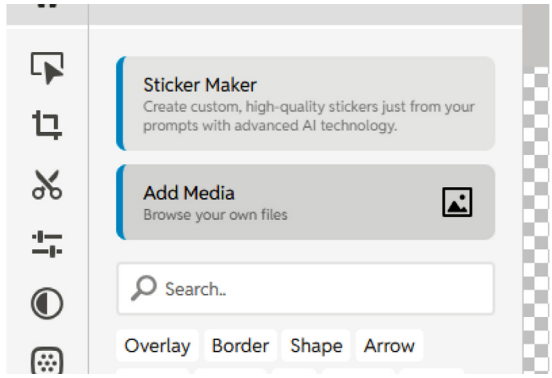


Figure 3.29 Element Option

Step 3: **Navigate** and **select** the **image** that you want to insert and click on **open**.

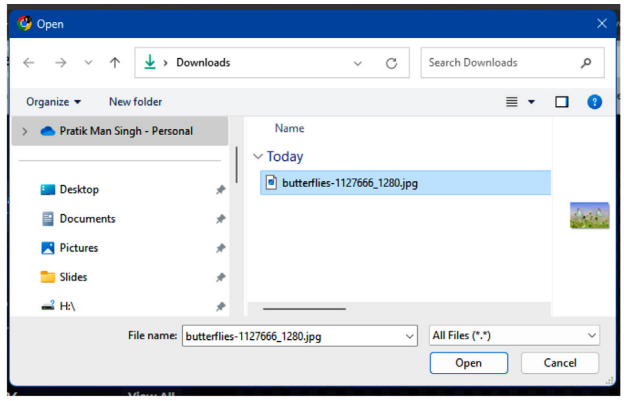


Figure 3.30 Explorer Window

Step 4: And the image will be inserted in the existing project.

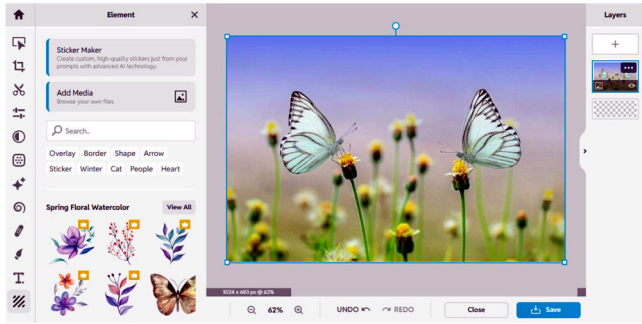


Figure 3.31 Pixlr X Interface

Using Cutout Tool

To further edit the graphics inside Pixlr X, you can use the Cutout tool. To utilize Cutout tool, you can follow the following steps:

Step 1: Open an image in Pixlr X and click on the Cutout icon on the left side of the screen.

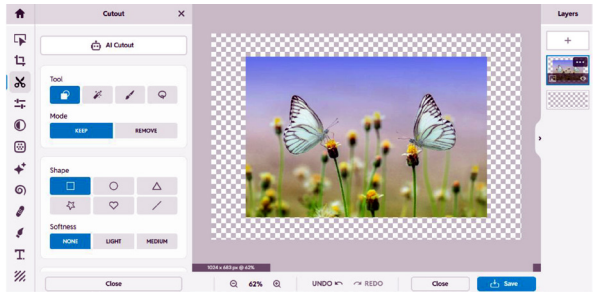


Figure 3.32 Pixlr Interface

Step 2: Choose the **Cutout** from the options on the left-hand side.

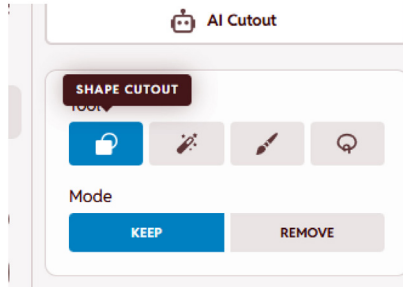


Figure 3.33 Cutout Option

Step 3: Choose the shape of the cutout according to your preference. And also select the softness level to adjust the level of softness in edges.

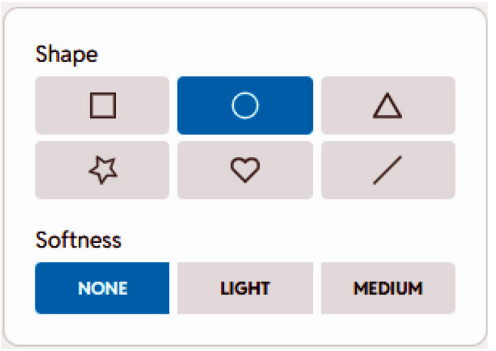


Figure 3.34 Layout Option

Step 4: Click and drag your mouse over the object you want to cut out.

Figure 3.35 Image Cutout

Figure 3.36 Image Cutout

Step 5: You can also use the Keep and Remove buttons to refine your selection. The selected area will be then added or removed according to the choice.

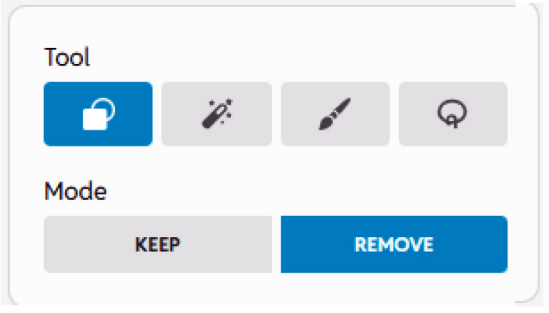


Figure 3.37 Cutout Tool

Figure 3.38 Image Cutout

Step 6: You can also utilize the Draw cutout tool to add or remove portions of image by freely drawing contents.

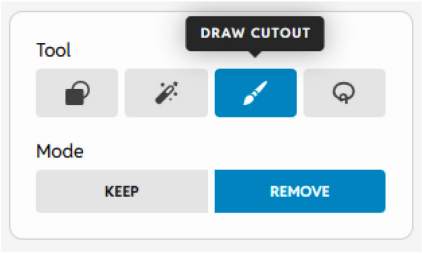


Figure 3.39 Cutout Tool

Figure 3.40 Image Cutout

Step 7: You can also utilize the Lasso cutout tool to add or remove portions of the image by freely drawing shapes.

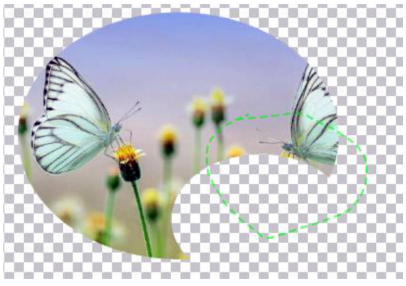


Figure 3.41 Image Cutout

Figure 3.42 Image Cutout

Exporting Project

Once the project is edited and ready to share it or to download it. You need to export the project. To export the project, the following steps needs to be followed:

Step 1: Click on the **Save button** at the bottom right corner of the Pixlr X interface.

Figure 3.43 Save Button

Step 2: This will open a pop-up window where you can choose the file name, format, and quality of your image.

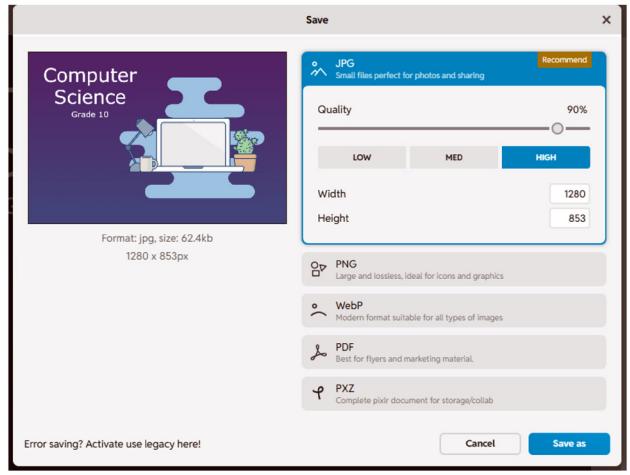


Figure 3.44 Save Window

Step 3: Choose the format that suits your needs. Pixlr X supports JPG, PNG, WEBP, and PXZ formats.

Note: JPG is a common format for web and print, PNG supports transparency, WEBP is a newer format that offers better compression, and PXZ is a layered Pixlr image format that preserves your edits and layers for future use.

Step 4: Adjust the quality slider to balance the file size and image quality. Higher quality means larger file size and sharper image, while lower quality means smaller file size and more pixelation.

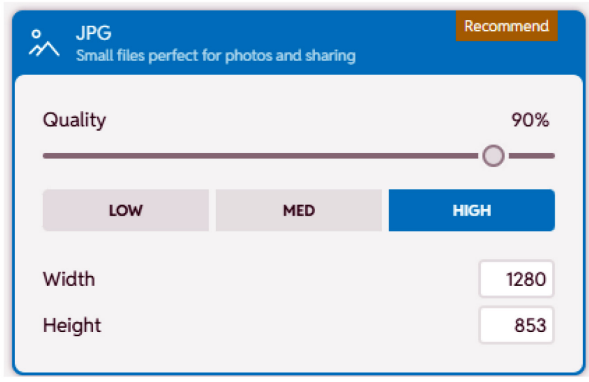


Figure 3.45 Quality Setting

Step 5: Click on the Save As button to save your image to your device.

Figure 3.46 Save As Botton

Step 6: Then, an explorer window pops up where you can choose the location of the file to be saved to and save the file.

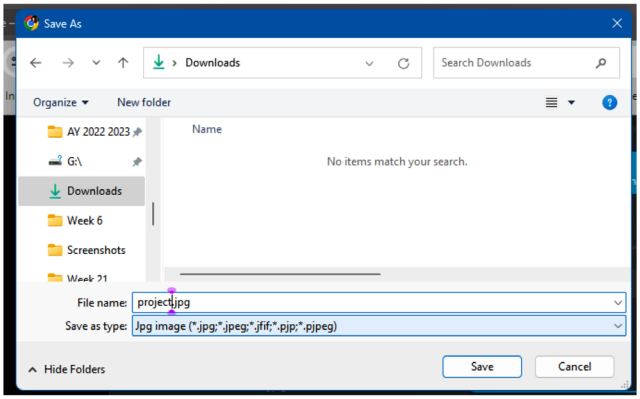


Figure 3.47 Explorer Window

Activity 1.1

Activity Outcome:

- Able to create and edit graphics using image editing tools.

Required resources: Computers/laptops, Internet connection, Pixlr X or Adobe Photoshop, sample images.

Procedure: Your teacher will guide you through the process of creating and editing graphics. Follow the steps below:

1. Open an image in Pixlr X or Adobe Photoshop.
2. Work with layers to separate elements such as background and text.
3. Use selection tools to edit specific areas of the image.
4. Add shapes, insert text, and apply filters or effects as needed.
5. Export the edited graphic in **PNG** or **JPG** format.

Result: Each student will present their edited image. The teacher will provide feedback and suggestions for improvement.

3.3 Audio

Audio refers to sound in multimedia. It may refer to the different sounds like music, voice, instruments or effects. These types of sounds are produced as a result of vibrations and in a computer device, these sound vibration data is stored as a digital file.

3.3.1 Audio Waveform

Sound changes over time; and to represent the change in sound, we use a graph which is known as an audio waveform.

The forms consist of Amplitude and Wavelength.

Here, Amplitude represents the loudness of a sound; and in the graph, it is represented by the height. And then, the Wavelength represents the nature of vibration of sound over time; and in the graph, it is represented by width.

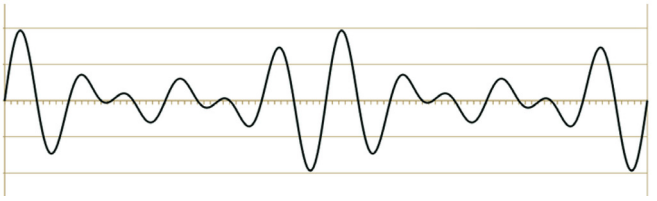


Figure 3.48 Audio Wave

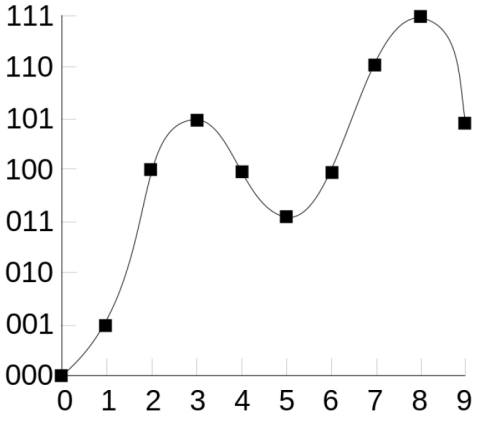


Figure 3.49 Audio Wave Sample

The graph can be different as per the nature of the sound. And to store them digitally, samples are collected in a certain time interval which is the digital representation of a waveform. This digital representation is stored as a sound file.

Common Audio File Formats

There are many audio files in use but the most common ones in use are MP3, AAC, and WAV.

MP3 - MP3 stands for MPEG Audio Layer 3. It is the most commonly used format for sharing audio files due to its effective compression capabilities.

AAC - AAC stands for Advanced Audio Coding. This is another commonly used format for sharing audio files, which is an improvement over MP3 as ACC uses more advanced compression algorithms.Due to this reason it has better compression capabilities compared to MP3 as well.

WAV - WAV stands for Waveform Audio File Format. It is an uncompressed audio file format due to which the filesize of the WAV audio file is large. However, the audio is stored exactly in the form as it is recorded.

Do you know?

MP3 revolutionised digital music by making audio files much smaller without significant quality loss, enabling the rise of portable music players and streaming services!

3.3.3 Audio Editing

In order to start editing audio, you can use Audiomass. Audiomass is a free online tool that lets you record and edit audio easily. You can also download and use an app version of it, named Audio Mass Audio Editor DAW, on your mobile phone. Using Audiomass is a great way to start editing audio.

Before starting to work in Audiomass, you need to know that there are three main areas of the application. They are the Menu bar, Waveform display and Control Panel that. You will be using in order to create and edit projects.

Figure 3.50 Audio Mass Interface

Menu Bar - Can be used to access various options such as file, edit, view, effects, and help.

Waveform Display - Can be used to see the audio waveform and perform basic editing operations such as cut, copy, paste, delete, and trim.

Control Panel - Can be used to play, pause, stop, record, and adjust the volume of the audio.

Recording Audio

Before you can start editing audio. You need to record audio. For that, you need to follow these steps:

Step 1: Go to the Audiomass website by visiting https://audiomass.co using your web browser. Once the website loads, you'll see the audio editing interface.

Step 2: Click on the **Record button** on the control panel, or press the **R** key on your keyboard.

Step 3: Now a **New Recording** dialogue box will appear.

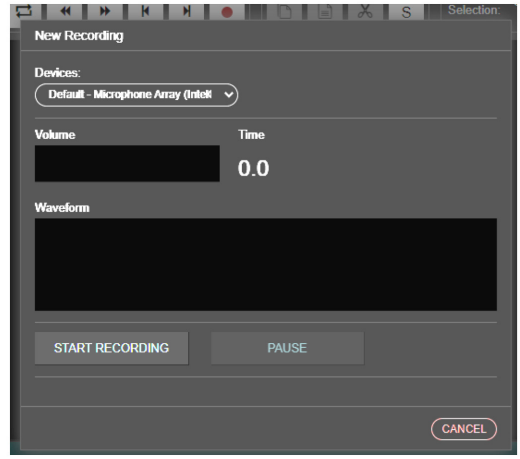


Figure 3.51 New Recording Dialogue Box

Step 4: In the devices section, make sure that the correct microphone is selected for recording from the Devices section.

Figure 3.52 Microphone Option

Step 5: And then, to start recording, click on the "Start Recording" button.

Figure 3.53 Start Recording Button

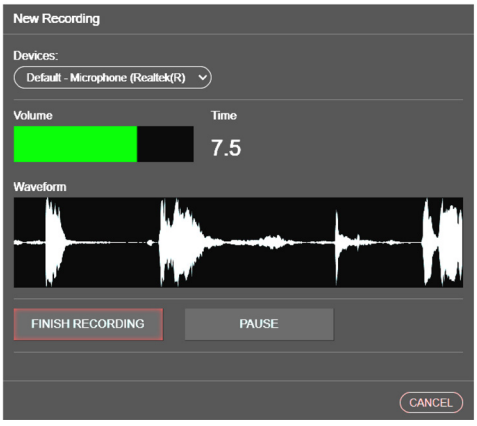


Figure 3.54 New Recording Diagolue Box

Step 6: This will start recording audio.

Step 7: Now, to end the recording, click on the "Finish Recording" button.

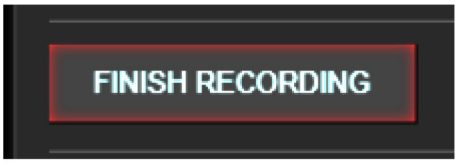


Figure 3.55 Finish Recording Button

Step 8: Then, click on the "Open Recording" button to open the recording in the waveform editor.

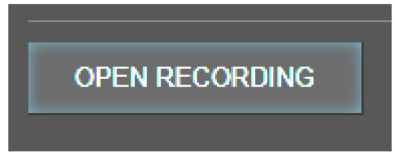


Figure 3.56 Open Recording Button

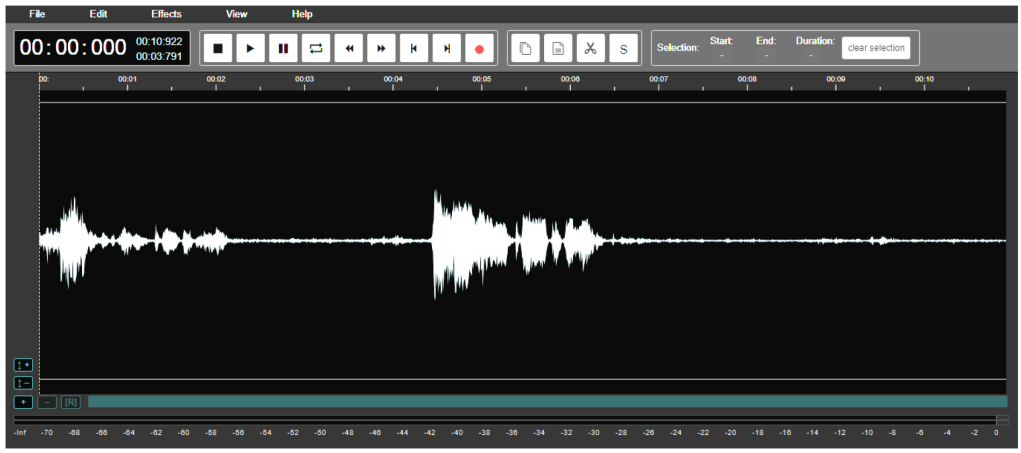


Figure 3.57 Audiomass Interface

Importing Audio

One of the ways to add an audio in Audiomass is by importing a **pre-existing** audio file. To import audio files, you need to follow these steps:

Step 1: Click on the File button on the top left corner and select "**Load from Computer**".

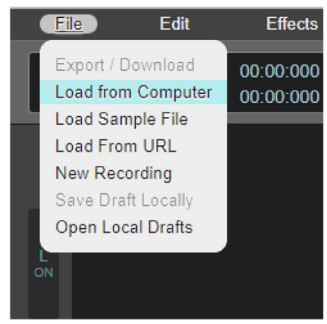


Figure 3.58 File Menu

Step 2: A file browser window will appear. Navigate to the folder where the audio file is saved and select it. Then, click on the **Open** button.

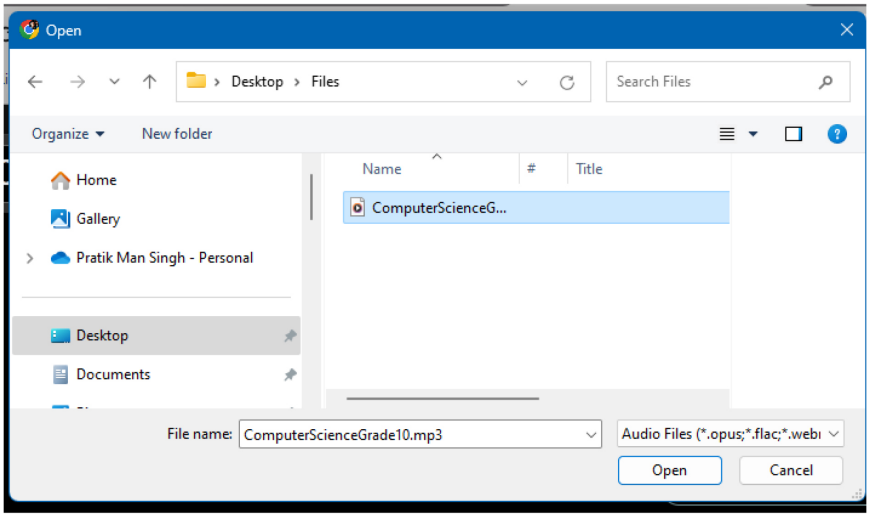


Figure 3.59 Explorer Window

Step 3: Audiomass will import the audio file and display it as a waveform in the editor window.

Figure 3.60 Audiomass Interface

Cutting Audio Waveform

Once audio has been recorded or imported in Audiomass, you may want to remove the audio that is unnecessary. To do so, you need to cut the audio and following these steps:

Step 1: Select the part of the waveform that you want to cut by clicking and dragging on it.

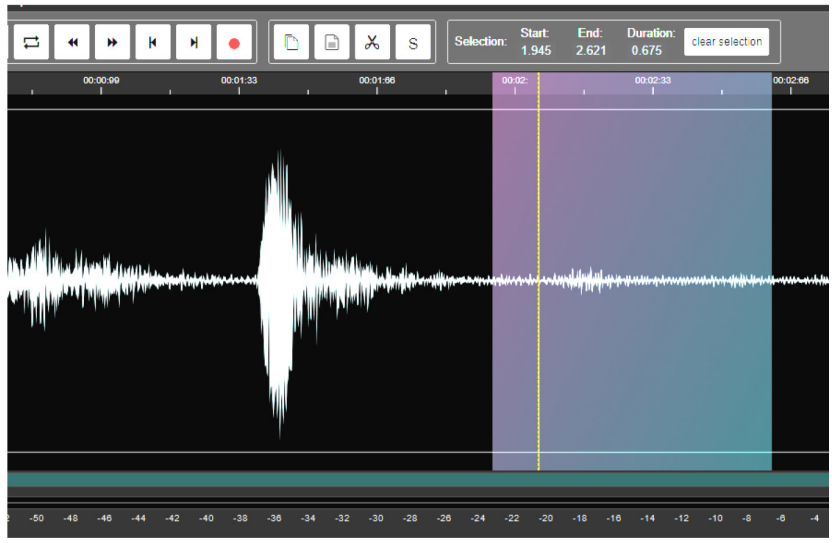


Figure 3.61 Audio Selection

Step 2: You will see a blue selection box around it.

Step 3: Click on the scissors icon on the toolbar to cut the selected part.

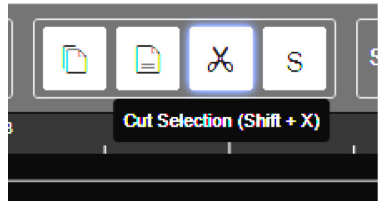


Figure 3.62 Tool Bar

Figure 3.63 Audiomass Interface

Applying Effects

Once you have the audio portion that you need, you can make its sound interesting by adding audio effects to it. To do so, you need to follow these steps:

Step 1: Select the portion of the audio that you want to apply the **effect** to.

Figure 3.64 Audio Selection

Step 2: You can do this by **clicking and dragging** on the **waveform.**

Step 3: Click on the "**Effects**" button on the menu bar. A list of available effects will appear, such as normalize, fade in, fade out, reverse, pitch shift, and more.

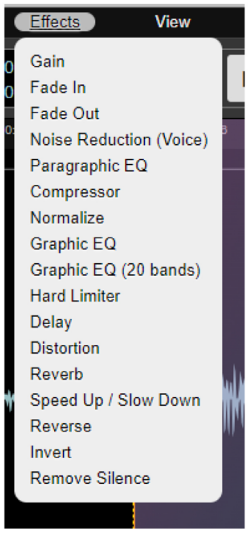


Figure 3.65 Effects Menu

Step 4: Choose the **effect** that you want to apply and **adjust the parameters** if needed.

Step 5: For example, if you want to apply a reverb **effect**, you can adjust the time decay and wet parameters and click on the **Apply** button to apply the effect.

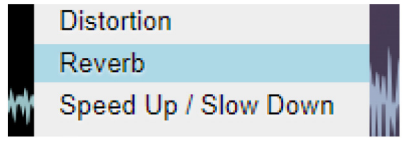


Figure 3.66 Reverb Option

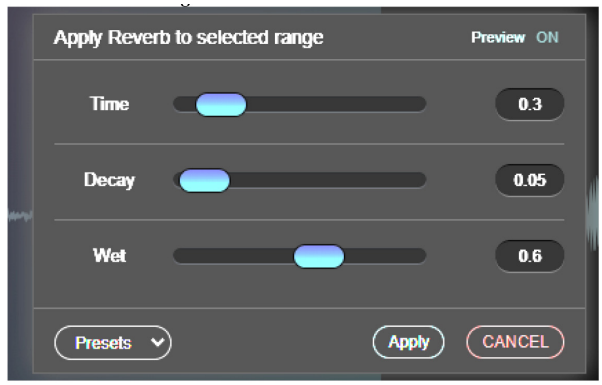


Figure 3.67 Reverb Dialogue Box

Step 6: Then, you can play the audio to hear how the applied effect sounds.

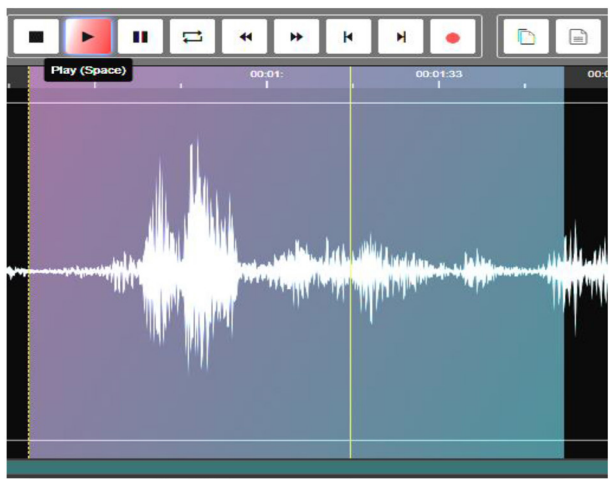


Figure 3.68 Audio Selection

Exporting Audio

Once the audio is edited as required, you can export it in various formats as you like. To do so, you can follow these steps.

Step 1: Click on the "**File**" button on the top left corner of the screen.

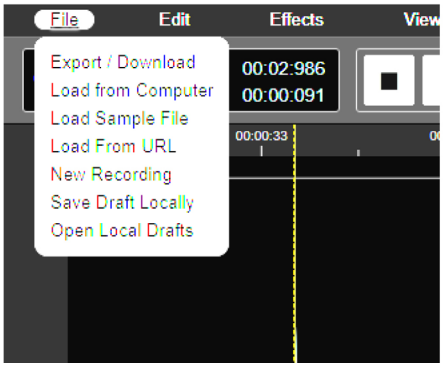


Figure 3.69 File Menu

Step 2: Click on the "Export/Download" option from the drop-down menu.

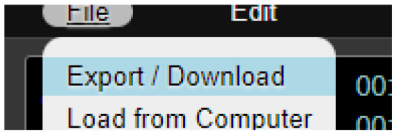


Figure 3.70 Export/Download Option

Step 3: Now a **dialogue box** will appear, where you can name the file, choose the format of audio as mp3 or wav, choose its **quality** (bitrate) and **audio channels.**

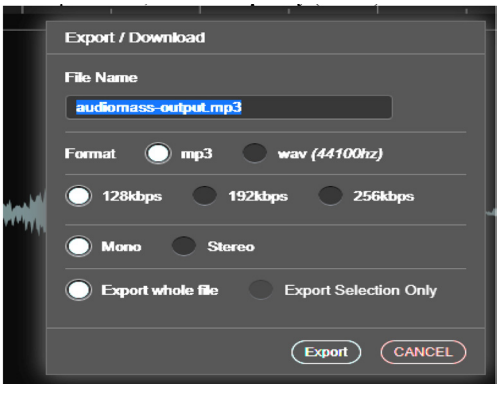


Figure 3.71 Export Download Dialogue Box

Step 4: Now, click on the "**Export**" button and the audio will be exported with selected settings.

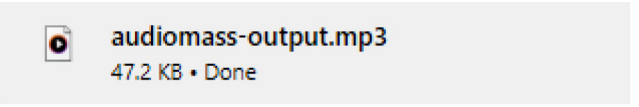


Figure 3.72 MP3 File

Activity 1.2

Activity Outcome:
- Able to create and edit audio using basic audio editing tools.

Required Resources:
Computers/laptops, headphones, microphones, audacity or AudioMass, sample audio files.

Procedure:
Your teacher will assist you in using audio editing tools. Follow the steps below:

1. Record a new audio clip or import an existing audio file into **Audacity** or **AudioMass**.
2. Cut unwanted sections and apply transitions or effects such as **echo**.
3. Adjust volume levels for consistency.
4. Export the final edited audio in **MP3** or **WAV** format.

Result:
Students will share their audio clips with the class. The teacher will provide feedback on editing quality and creativity.

3.4 Video

Video in multimedia refers to a collection of pictures known as frames that runs over a certain unit of time to create the illusion of motion to the viewers. It also contains an audio file that plays along with the frames.

3.4.1 Frame Rate

The series of frames that runs or plays in a certain time interval is known as Frame Rate which can be measured as frames per second (FPS). The amount of frames per second determines whether a video is smooth or choppy. The higher the frame rate the smoother the video and vice versa.

Some common frame rates that are in use are typically 24fps, 25fps, 50fps and 60fps.

3.4.2 Resolution

Pixels are the tiny squares that make up a digital image. Each pixel has a color and all the pixels together form the picture. The more pixels an image has, the more details it can show.

Resolution is the number of pixels in an image. It is usually expressed as width and height in pixels, such as 720 × 480 or 1920x1080. Resolution affects the quality and size of an image. A higher resolution means a clearer and larger image, but also a bigger file size.

Common resolutions are standards that are used for different purposes, such as displaying images on screens or printing them on paper. Some common resolutions are:

SD (Standard Definition): 720 × 480 pixels. This is a low resolution that is used for older TVs and DVDs. It can show basic colors and shapes, but not much detail.

HD (High Definition): 1280 × 720 pixels. This is a medium resolution that is used for newer TVs and online videos. It can show more details and colors than SD, but not as much as FHD.

FHD (Full High Definition): 1920 × 1080 pixels. This is a high resolution that is used for newer TVs, monitors, and laptops. It can show a lot of details and colors, and it is the most common resolution for HD videos and games.

Activity 1.3

Activity Outcome:

- Able to identify and differentiate SD, HD, and FHD resolutions based on quality, size, and usage.

Required Resources:

Computers/laptop and sample video clips in SD (720×480), HD (1280×720), and FHD (1920×1080)

Procedure:

Your teacher will assist you in understanding video resolution through a comparative activity. Follow the steps below:

1. Watch three sample videos shown in SD, HD, and FHD resolutions.
2. Observe carefully the sharpness, clarity, and level of detail in each resolution.
3. Complete the comparison table below based on your observations.
4. In groups or pairs, discuss the advantages and limitations of each resolution type.
5. Answer the reflection questions based on cases of real-life use and technical understanding.

Comparison Table

Resolution	Dimensions (W × H)	Picture Quality	File Size	Common Usage Examples
SD	720 × 480			
HD	1280 × 720			
FHD	1920 × 1080			

Reflection Questions (Answer briefly):

1. Which resolution displayed the most visual clarity and details? Why?
2. If you were uploading a school project video online, which resolution would you choose and why?
3. Which resolution is best suited for mobile viewing with limited internet data?
4. Why do movies and modern games prefer FHD or higher resolutions?

Result: Each student or group will submit the filled worksheet to the teacher. The teacher will review, observation accuracy, explanation clarity and logical reasoning in answers.

Students will gain practical understanding of video resolution and its impact on multimedia quality.

3.4.3 Video File Formats

Among the many video file formats in use, MP4, MOV and AVI are the most commonly used video formats.

i. MP4 - MP4 stands for MPEG-4 Part 14. This is the most commonly used video format due to its high level of compression capabilities

ii. MOV - MOV is an extension of QuickTime Movie file format. It is commonly used to store videos which may be compressed or uncompressed as per the requirement.

iii. AVI - AVI stands for Audio Video Interleave. This video format is also commonly used to store videos which may be compressed or uncompressed as per the requirement.

Do you know?

MP4 is the most widely used video format because it balances high-quality visuals with efficient compression, making it ideal for streaming platforms like YouTube and Netflix!

3.4.4 Video Editing

In order to start editing videos, you can use CapCut. You can also download and use an app version of it, named CapCut - Video Editor, on your mobile phone. CapCut is a free online tool that lets you edit videos easily. You can cut, arrange, add transitions, add effects and more. CapCut is a great way to start learning about video editing.

Before starting to work in CapCut, you need to know that there are four main areas of the application. They are the **Timeline Area, Tools Panel, Preview Area and Export and Share Option** that you will be using in order to create and edit projects.

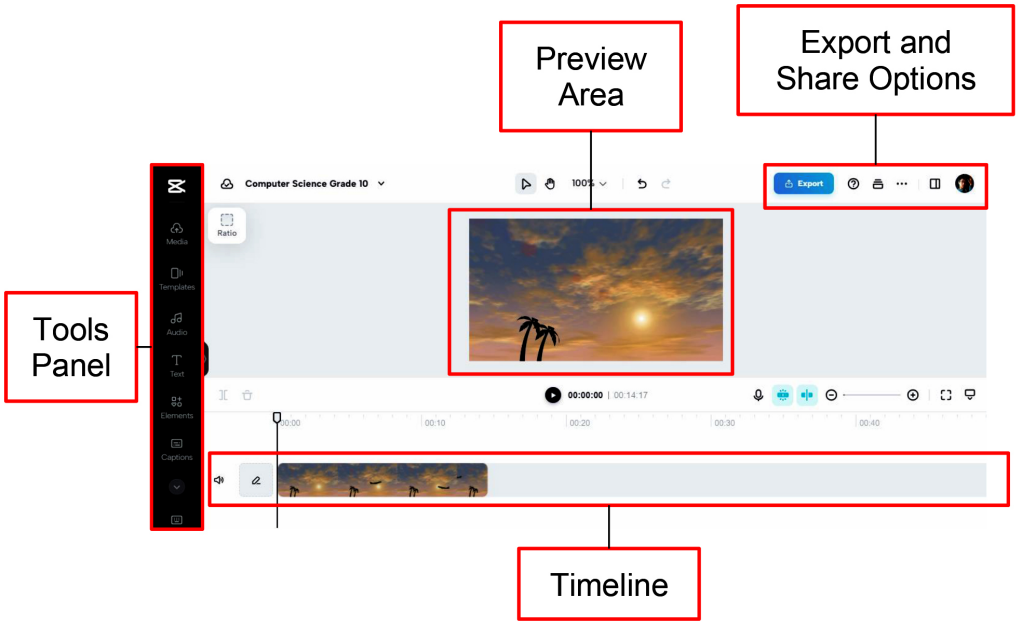


Figure 3.73 Capcut Interface

Timeline Area: This area is where you arrange and edit your video clips, audio, effects, and text.

Tools Panel: This is the editing tools section from which you can access videos, audio, transitions, effects and other editing options.

Preview Area: This is where you can see the preview of the project being edited.

Export and Share Options: This is from where you can access options to export the edited videos in different formats and share your videos with others.

Creating a New Project

Before you can start editing videos in CapCut, you need to create a new CapCut Project. The steps to do so are as follows:

Step 1: Go to CapCut's website (www.capcut.com).

Step 2: From the Homepage click on **"Create new video"** button.

Figure 3.74 Create New Video Buton

Step 3: Now, the **"Select aspect ratio"** dialogue box pops up where you need to choose an aspect ratio of your choice. For example, you can choose a standard **"16:9"** Aspect Ratio.

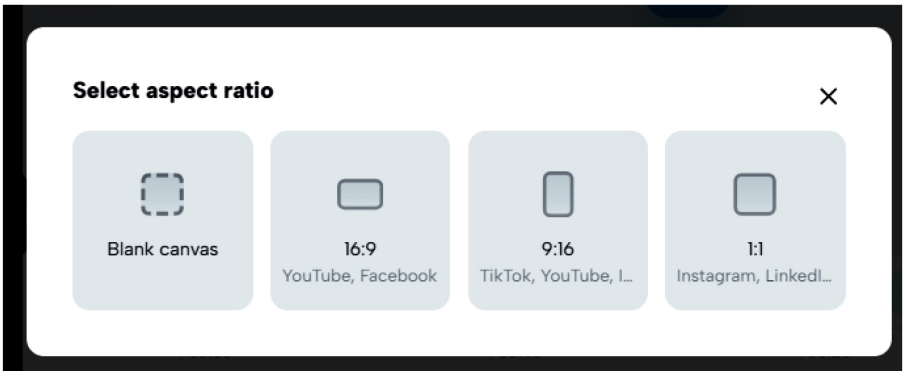


Figure 3.75 Aspect Ratio Selection Window

Step 4: Then, a new project will be created. And you can see the main editing interface.

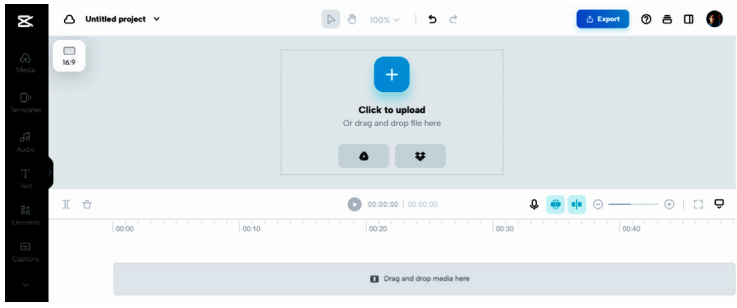


Figure 3.76 Editing Interface

Step 5: On the top left hand corner, you will be able to see "**Untitled project**", you can click on it and **rename** the project as you like to keep the name of the project.

Figure 3.77 Title Bar

Step 6: After that, the project will be renamed.

Importing Videos

Once you are in the editing interface, you need to import videos and other media that you want to edit. For that, you can follow these steps:

Step 1: Click on "**Media**" on the left hand side of the interface and choose the project.

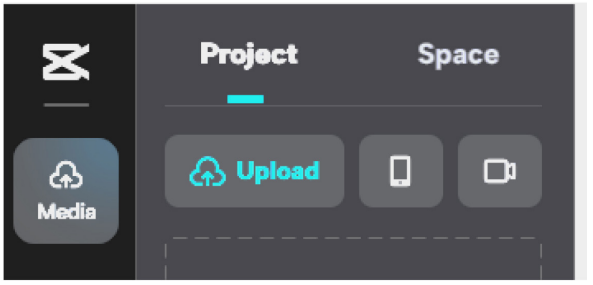


Figure 3.78 Media Section

Step 2: After that, click on the **Upload** button and options for upload will pop up.

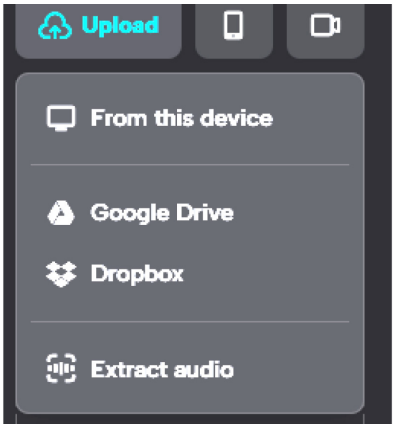


Figure 3.79 Upload Option

Step 3: Now, in order to upload media files stored on your computer, click on "**From this device**" and an explore window will appear.

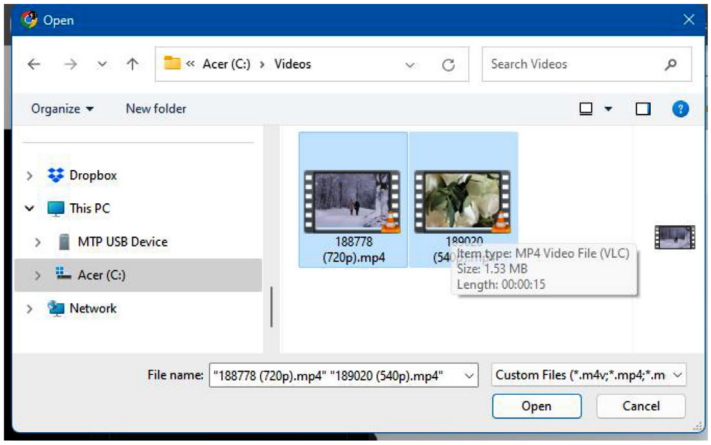


Figure 3.80 Explorer Window

Step 4: Now, you need to choose the video or media files that you want to import into the project and click on the "**Open**" button.

Step 5: After that, the videos or media files will be imported into the **Project section**.from where you can access them for editing.

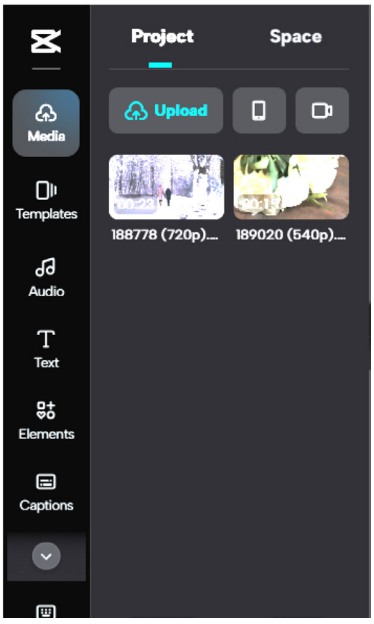


Figure 3.81 Project Section

Working in Timeline

Once the project has been set up. To start editing videos, you need to arrange the videos or other media elements in the timeline. To do so, you have to follow these steps:

Step 1: Click on the media or video file in the **project section** and it will be added into the **timeline**.

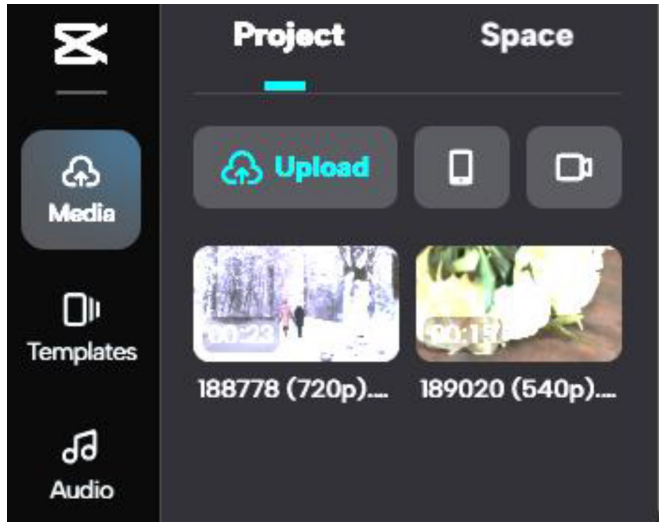


Figure 3.82 Project Section

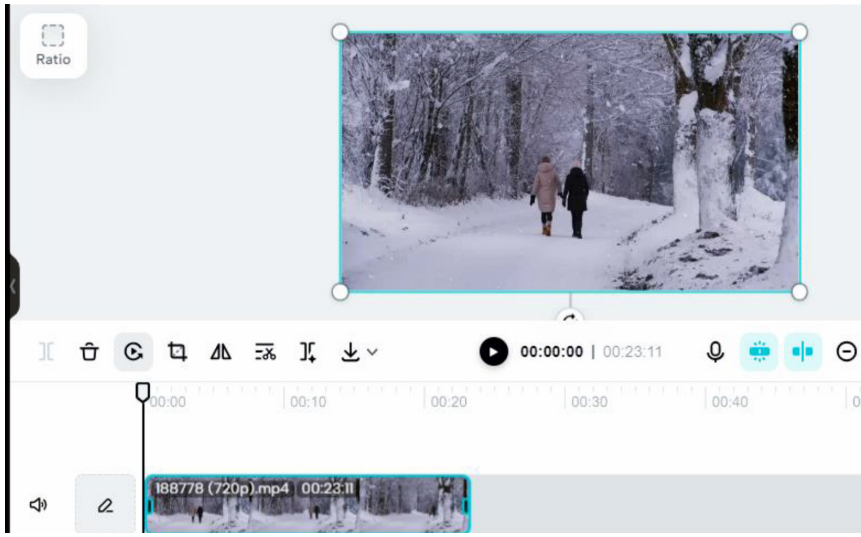


Figure 3.83 Editing Area

Step 2: You can also alternatively **click and drag** the media or video file from the **project panel** into the timeline.

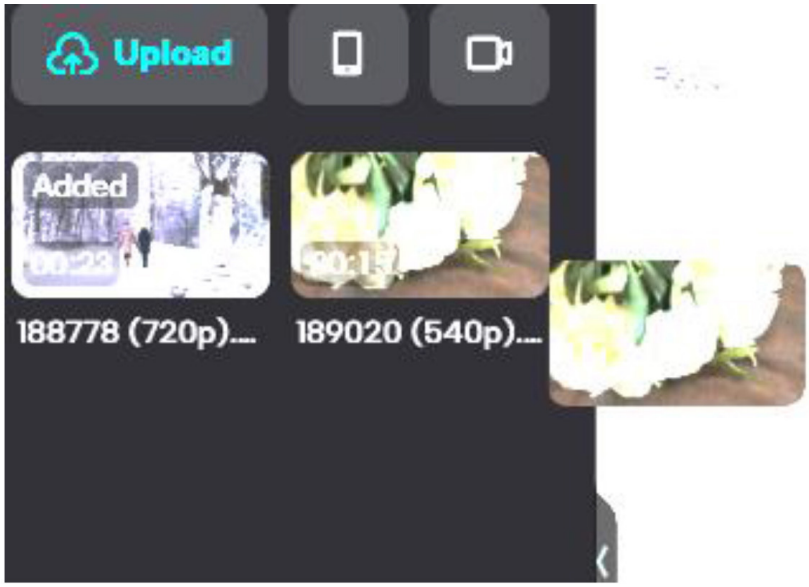


Figure 3.84 Project Section

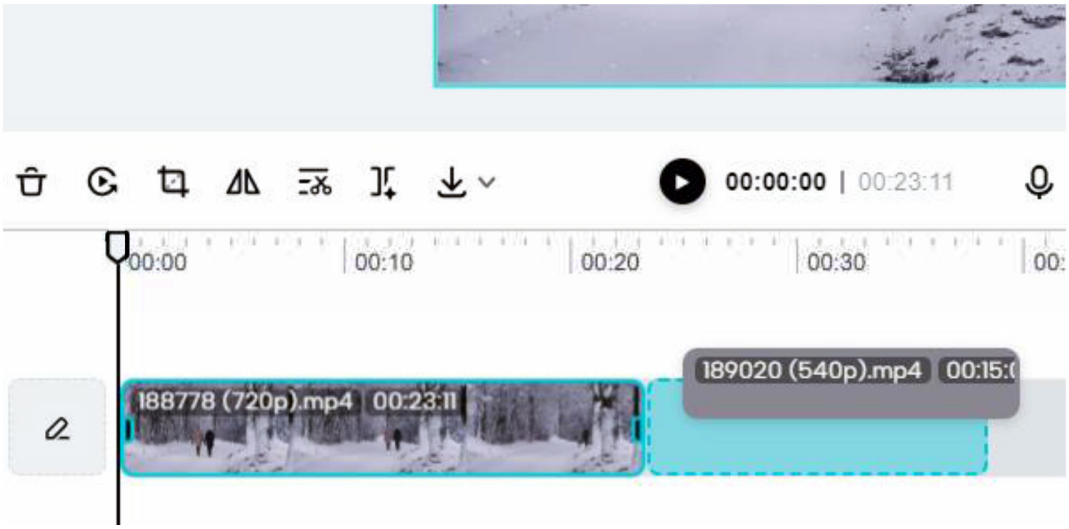


Figure 3.85 Timeline

Step 3: The videos will be arranged in the **timeline** as per the order that you place them.

Figure 3.86 Editing Area

Step 4: To rearrange the order of the clips, tap and hold on a clip and drag it to a **new position** on the timeline.

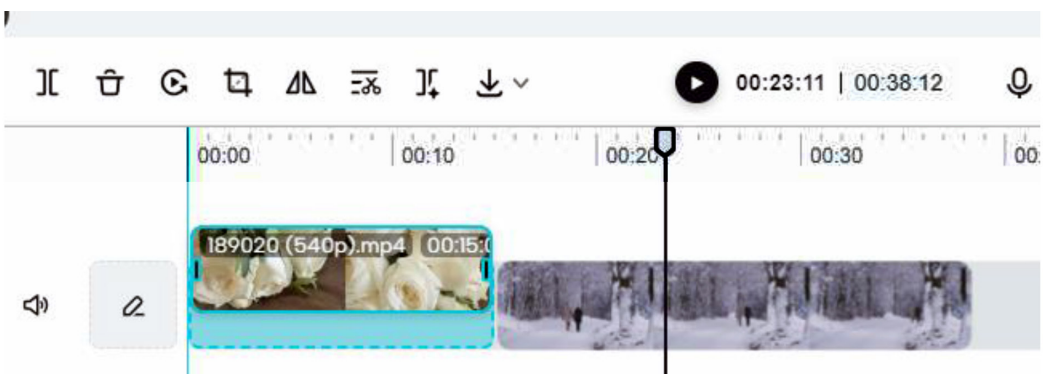


Figure 3.87 Timeline

Step 5: You can also **zoom in or out** of the timeline using the **zoom slider.**

Step 6: To work with the timeline cursor, tap on the play button in the preview section or press the "**SpaceBar**" on your keyboard.

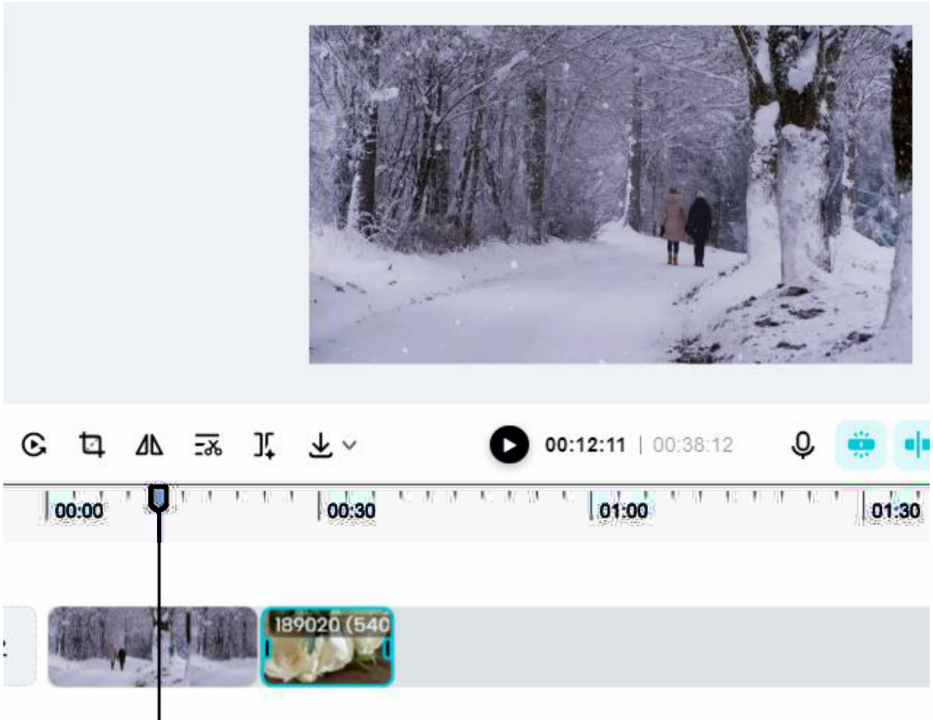


Figure 3.88 Editing Area

Step 7: The cursor will move along the **timeline** and show you the **preview** of your video.

Step 8: You can **pause** or **resume** the playback with the same button.

Step 9: You can also **drag the cursor** to any position on the timeline to jump to that point in the video.

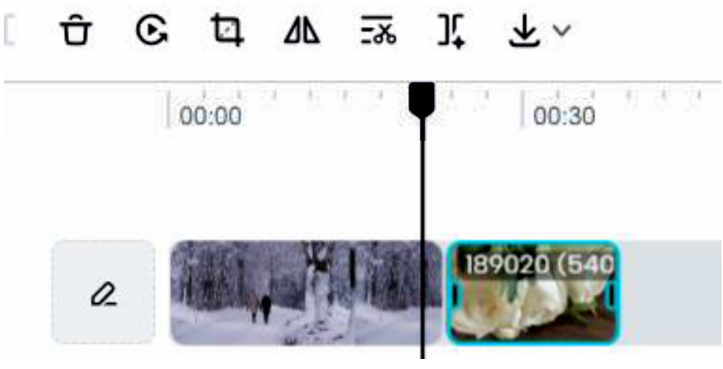


Figure 3.89 Timeline

Step 10: The current time of the cursor is displayed on the button left hand side of the preview window.

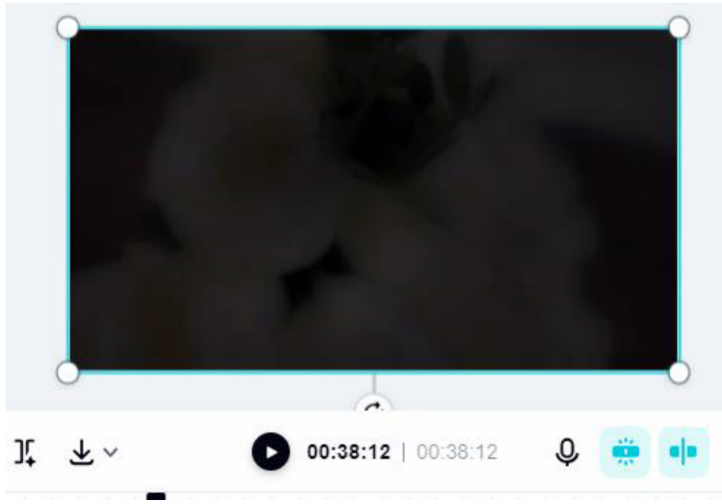


Figure 3.90 Preview Area

Trimming and Cutting Videos

Step 1: To trim the video in the timeline, drag the handles at the beginning or end of the video clip to adjust the duration.

Figure 3.91 Timeline

Step 2: To cut the video, move the play head to the position where you want to split the video and click on the Split button.

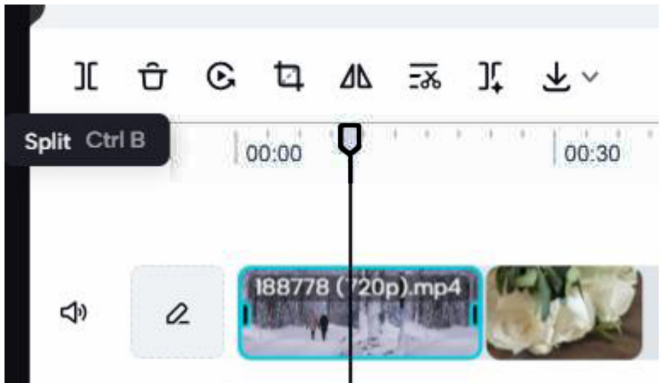


Figure 3.92 Timeline

Figure 3.93 Timeline

Step 3: You can delete the unwanted parts of the video by selecting them and clicking on the Delete button.

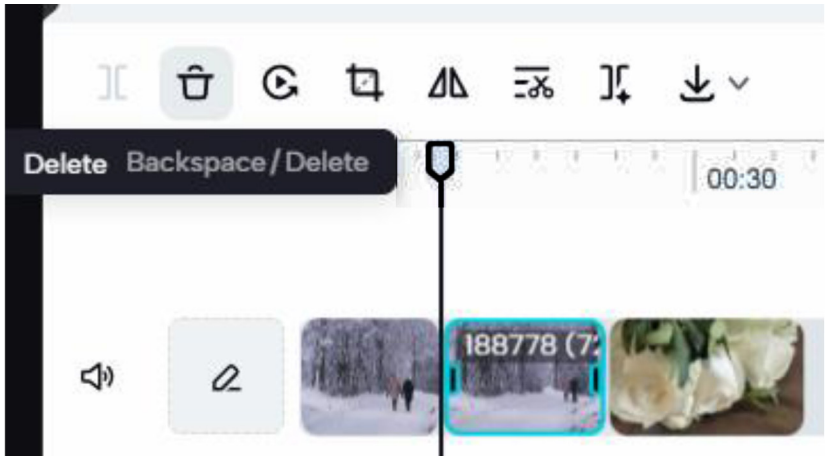


Figure 3.94 Timeline

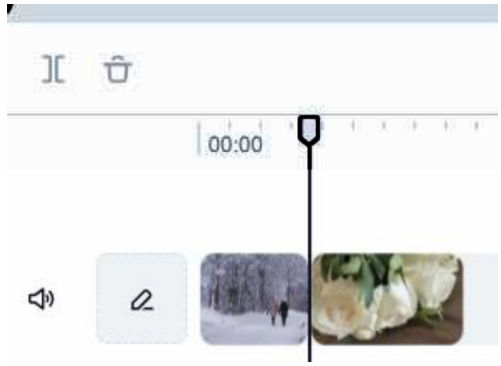


Figure 3.95 Timeline

Adding Transition

Once you arrange and trim your video clips, you can add transitions in between video clips where necessary. The process of adding transitions are as follows.

Step 1: To add transitions between video clips, click on the **"Add Transition"** button that appears when you hover your mouse between two clips.

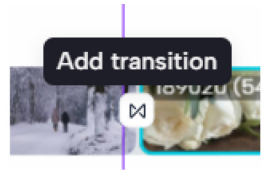


Figure 3.96 Transition

Step 2: Now, the Transition **Menu** will appear on the left side that displays various available transitions.

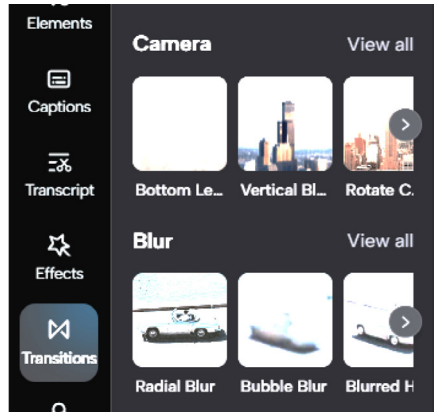


Figure 3.97 Transition Option

Step 3: Browse through them or use the search bar to find a specific transition.

Step 4: Click on the desired transition to apply it.

Step 5: You can also adjust the duration of the transition by dragging the edges of the slider.

Figure 3.98 Transition

Step 6: Repeat the steps for adding more transitions between other clips.

Adding Effects

To make the video clips look interesting, various video effects can be added as well. The process of adding video effects are as follows:

Step 1: Click on the "**Effects**" button at the left panel to open the effects menu.

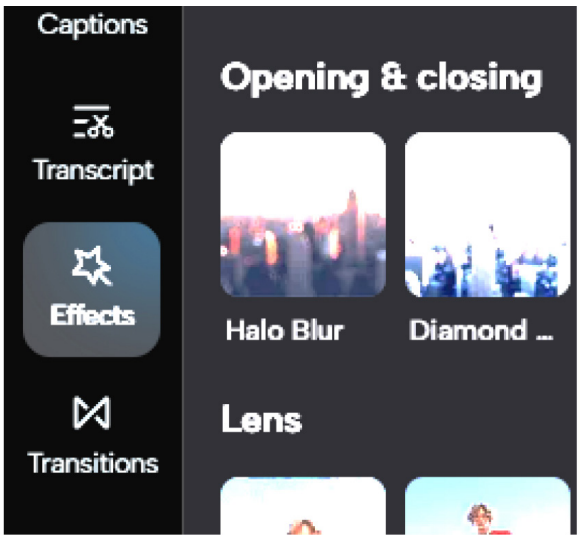


Figure 3.99 Effects Option

Step 2: This menu will show you different categories of effects.

Step 3: You can browse through them to find a specific effect.

Step 4: Make sure that the timeline cursor is in the starting position of where the effect is to be applied from.

Figure 3.100 Timeline

Step 5: Now, click on the desired effect to apply it.

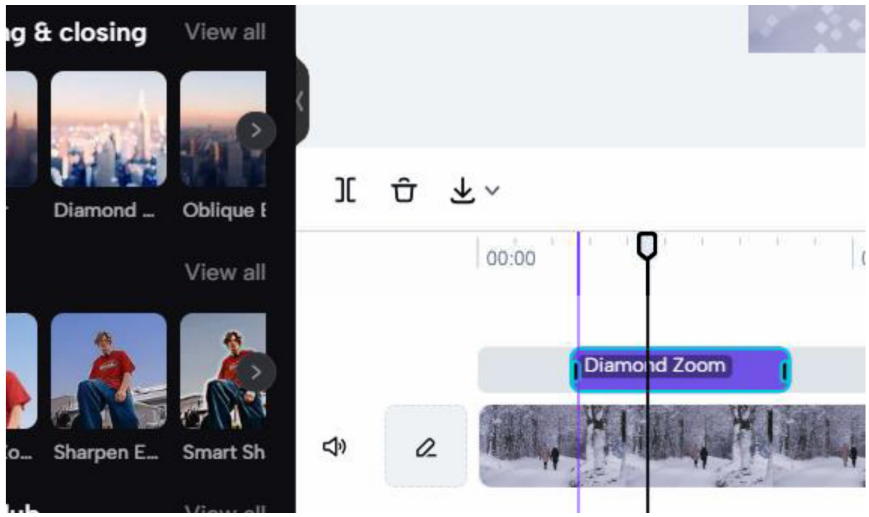


Figure 3.101 Effects

Step 6: To get the specific effect option, you can double click on the Effects applied on the timeline.

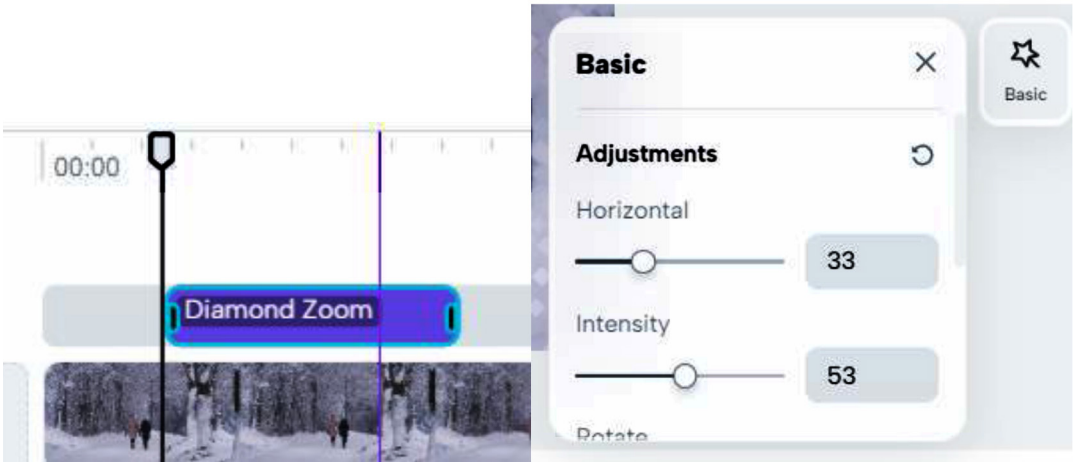


Figure 3.102 Effect Option

Step 7: Now, adjust the properties of the effects to change its behaviour.

Step 8: Repeat the steps for adding more effects to other clips.

Adding Text

You can add different elements such as **Text** in video projects. The steps to add text in CapCut video projects are as follows:

Step 1: Click on the **Text icon** on the left side of the interface.

Figure 3.103 Text Tool

Step 2: From here, you can choose to add basic text or choose a text template.

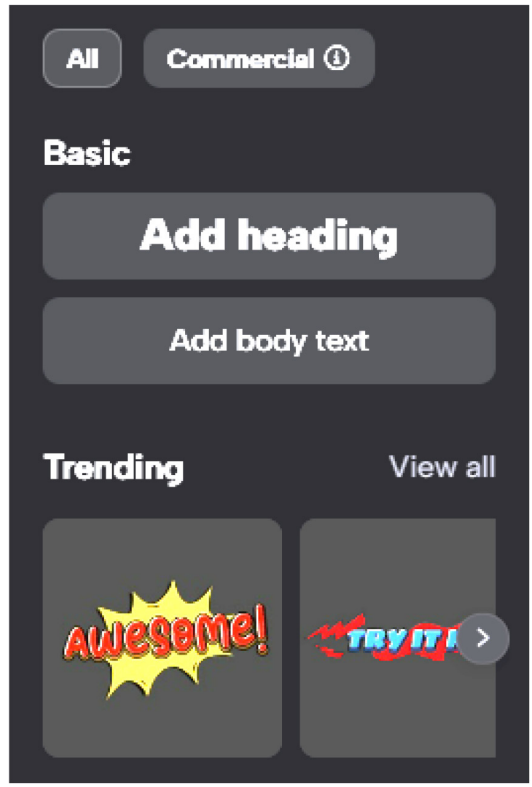


Figure 3.104 Text Option

Step 3: For example, you can click on **"Add heading"** and a text box will appear on the video preview.

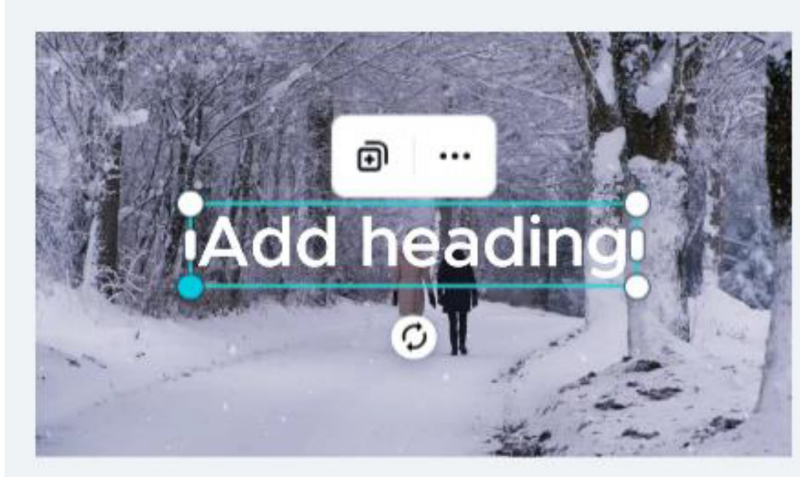


Figure 3.105 Text Box

Step 4: Here, you can double click on the text box and type in whatever you want.

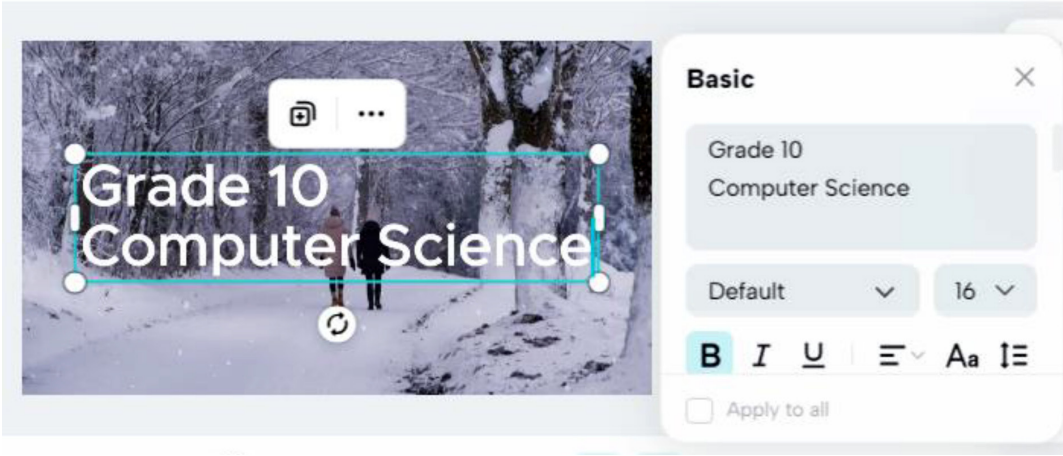


Figure 3.106 Text Option

Step 5: Once you do that, you can also change the text properties as you like from the Properties window that appears.

Step 6: To change the text duration, you can drag the edges of the text clip to adjust its start and end time.

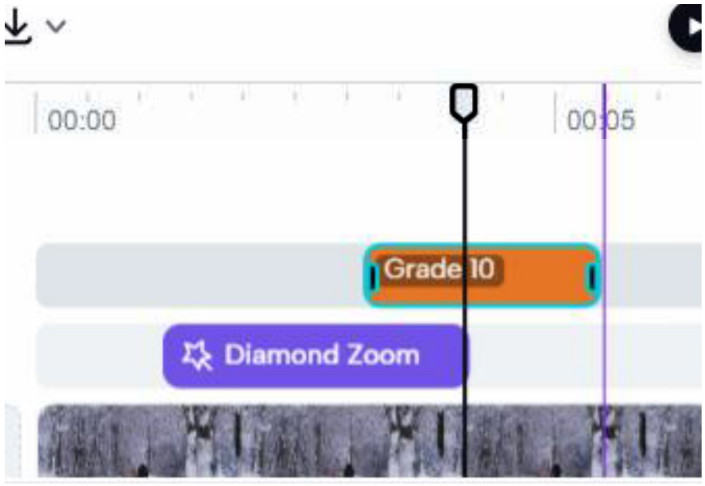


Figure 3.107 Timeline

Exporting Project

After the editing process is over, you can export the video in required format. The process to do so is as follows:

Step 1: Make sure that the project that you want to export is open in CapCut.

Step 2: Click on the **Export button** at the top right hand side of the screen.

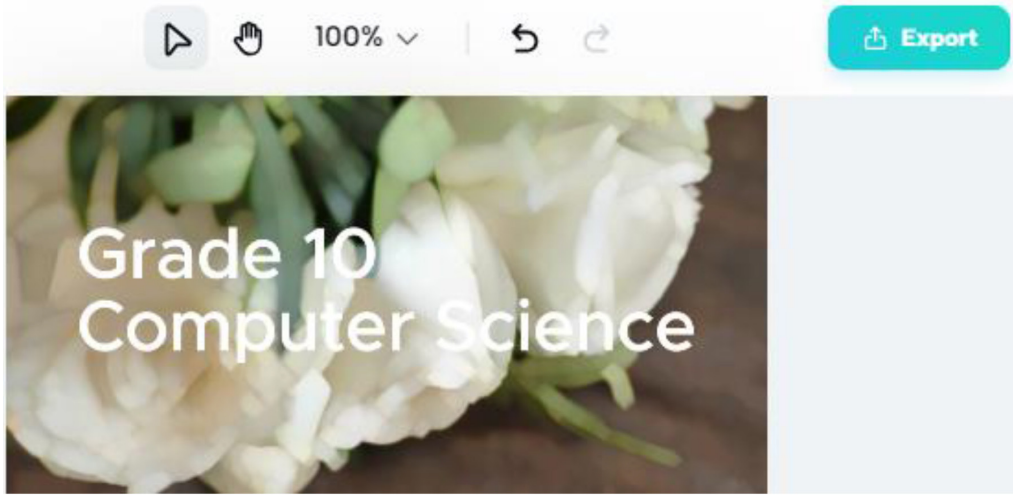


Figure 3.108 Export Option

Step 3: Now, a window opens in which you should click on the "**Download**" Button.

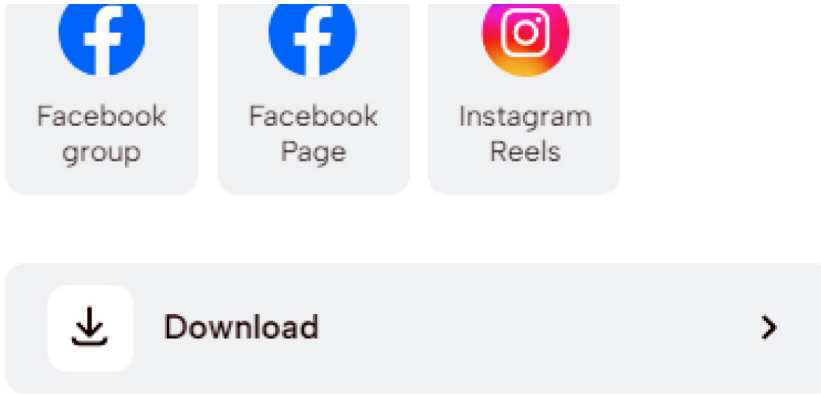


Figure 3.109 Download Option

Step 4: Select the quality, resolution and framerate for the video and click on the **export** button.

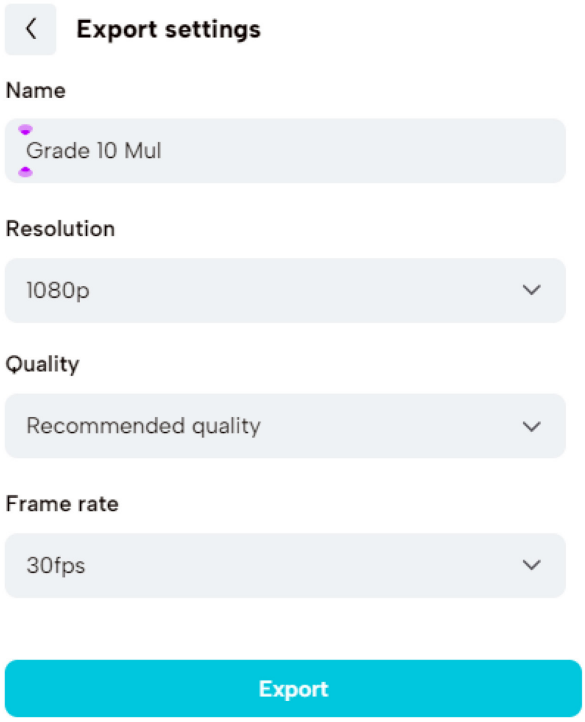


Figure 3.110 Export Settings

Step 5: Once the process of export is over, you will see the **Download** button. Click on it to download the video to your computer.

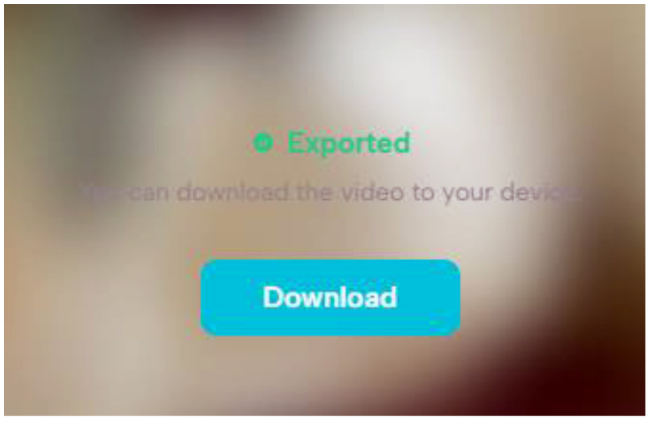


Figure 3.111 Download Option

Activity 1.4

Activity Outcome:
- Able to create and edit video using basic video editing tools.

Required Resources:
Computers/laptops, Clipchamp or CapCut, sample video clips, headphones.

Procedure:
Your teacher will guide you in editing a short video. Follow the steps below:

1. Import video clips into **Clipchamp** or **CapCut** and arrange them on the timeline.
2. Trim unnecessary sections and add transitions between clips.
3. Insert text overlays and apply filters or effects to enhance the video.
4. Export the final video in **MP4** format for sharing.

Result:
Each group or individual will present their video. Teachers will give feedback focusing on creativity, flow, and technical quality.

3.5 Concept of Animation

Animation refers to movements of pictures or graphics. In multimedia, animation refers to computer generated movements. Such animations can be used for creating cartoons, advertisements, movies and video games, etc.

Psychology plays a vital role in animation as well, in animation it means using ideas about how people think and feel to make animations more interesting. For example, using bright colours and moving objects can grab the viewer's attention quickly which is especially used in cartoons to capture children's attention and exactly the reason for why they are drawn into it.

Another example is funny or emotional faces on animated characters that help people connect with the story. Smooth and well-timed movements also make animations more enjoyable to watch. By understanding what attracts people's attention, animators can create animations that are more fun, exciting, and easier to understand.

Animation can be majorly divided into 2 types, 2D Animation and 3D Animation.

3.5.1 2D Animation Concept

2D animations are flat in nature as they are made in two dimensions. It means they have width and height only. 2D animation is usually drawn frame by frame in an animation software while making it.

3.5.2 3D Animation Concept

3D animations have perspective in them as they are made with three dimensions. It means they have width, height and depth. 3D animation is usually modelled, rigged and then movements are added in them.

Activity 1.5

Activity Outcome:
● Able to identify and classify multimedia components such as text, graphics, audio, video, and animation.

Required Resources:
Multimedia presentation or website, notebooks or worksheets.

Procedure:
Your teacher will show you a multimedia presentation or a website. Follow the steps below:

1. Observe the content carefully and identify the multimedia components used like **Text, Graphics, Audio, Video, and Animation**.
2. Note down examples of each component present in the content.
3. Participate in a class discussion on how these components enhance the overall quality and engagement of the presentation.

Result:
Class discussion led by the teacher with student reflections and feedback on how multimedia elements contribute to effective communication.

Do you know?

The first 3D animated feature film was Pixar's "Toy Story," released in 1995.

Exercise

1. Write the full forms of the following abbreviations.

i) RTF ii) PNG iii) GIF

iv) JPG v) SVG vi) PDF

vii) MP3 viii) AAC ix) MP4

x) AVI

2. Choose the correct answer.

i. What does the term "Multimedia" refer to?

a. Single type of media
b. Many types of media
c. A specific type of technology
d. Media manipulation

ii. Which is a drawback of multimedia according to the provided content?

a. Costly and inaccessible hardware/software
b. Capturing audience attention effectively
c. Enhancing personal communication
d. Creating immersive experiences

iii. Which of the following is a standard text file format that only stores text contents?

a. RTF c. DOCX
b. TXT d. PDF

iv. Which term is used to describe characters that we see on the screen and is also known as alpha-numeric values?

a. Multimedia c. Text
b. Notepad d. RTF

v. **What type of computer graphics is made up of bezier curves and maintains quality regardless of its size?**

a. Raster Graphics
b. Vector Graphics
c. PDF
d. GIF

vi. **Which file format is commonly used for sharing documents online and can contain both raster and vector graphics?**

a. JPG
b. PDF
c. SVG
d. GIF

vii. **What does Amplitude represent in an audio waveform graph?**

a. Sound wavelength
b. Sound compression
c. Sound loudness
d. Sound file format

viii. **Which audio file format is known for its effective compression capabilities and is commonly used for sharing audio files?**

a. WAV
b. AAC
c. MP3
d. FLAC

ix. **What does Resolution in video refer to?**

a. The number of frames per second
b. The amount of audio compression in a video
c. The number of pixels in an image, expressed as width and height
d. The series of frames played in a video

x. **Which video format is commonly used for storing videos with high compression capabilities?**

a. AVI
b. MP4
c. MOV
d. MKV

xi. **In 3D animation, what are the three dimensions that contribute to the perspective of the animations?**

a. Width, height, and depth
b. Width, length, and size
c. Height, ruler, and breadth
d. Length and width

xii. **What is the primary characteristic of 2D animations mentioned in the content?**

a. They are modeled and rigged.
b. They are created with three dimensions.
c. They have width, height, and depth.
d. They are drawn frame by frame in animation software.

3. Write short notes on:

a. Components of multimedia
b. Rich Text Format
c. Audio Waveform
d. Video Resolution and Framerate
e. Raster and Vector Graphics
f. 2D and 3D Animation

4. Answer the following questions.

a) What are the five main components of multimedia?
b) Define the term 'raster graphics'.
c) Name three common raster image file formats.
d) Explain the difference between raster and vector graphics.
e) Why is multimedia important in education and business?
f) How does a DOCX file differ from a TXT file?
g) If you have to create a greeting card, which multimedia components will you use, and why?
h) Suggest an appropriate image format for a transparent company logo and explain the reason for it.
i) Which type of graphic, raster or vector, would you use for designing a billboard? Justify your answer.

j) Compare the use of multimedia in platforms, YouTube and Google Earth. Explain the similarities and differences you can identify.

k) Draw a simple multimedia learning app interface and list which multimedia elements you would include with reason.

l) Evaluate the impact of video content in comparison to other media forms (e.g., text or audio) in digital communication.

5. Activities

1. Prepare a presentation file on "Importance of Multimedia and its Uses" and present it to your class as a group work.
2. Collect the names, logo, application areas, file extensions and other related information of different graphics, audio and video software and draw in a sheet of chart paper.

6. Project Work

Create and present a multimedia file using graphic, audio, or video editing Tools

Objective:

To demonstrate students' understanding of multimedia creation by producing a **graphic** or **video presentation**. This project encourages creativity while teaching practical skills in media editing and presentation.

Project Structure:

You will choose **one type of multimedia project** and complete the following steps:

1. **Graphic project:**
   a. Create a 3 promotional image or poster using graphic design tools.
   b. Save the final images in **.png** or **.jpg** format.

2. **Video project:**
   a. Create a video presentation using video editing software.
   b. Export the video in **.mp4** format.

Steps to Complete the Project:

1. **Choose a project type**

   Select **one** from the following:

   - 3 Promotional images (Graphic)
   - Video presentation

2. **Plan your content**

   - Define the **main message** you want to share.
   - Identify your **target audience**.
   - Sketch or write a basic script/storyboard for your content.

3. **Edit and refine**

   - Use appropriate tools to create and enhance your project.
     - Graphic tools: Pixlr X, Adobe Photoshop, etc.
     - Video tools: CapCut, Clipchamp, etc.
   - Make sure the final product is clear, creative, and engaging.

4. **Finalize and export**

   - Review and polish your work.
   - Export the project in the correct format:
     - .png /.jpg for image
     - .mp4 for video

5. **Presentation**

   - Present your multimedia project to the class.
   - Explain:
     - Tools you used
     - Techniques you applied
     - Message behind your project

Unit 4 Programming in Python

Let's solve!

What will be the output of the following Python codes:

1.
```
x = 3
y = 2
print ('Sum of', x, 'and', y, 'is', x + y)
```

2.
```
days int (input ("Enter days: "))
months days / 30
days days % 30
print ("Months = ", int (months), "Days = ", days)
```

In the previous class, you have learned what is Python, its features, how to install Python. To write and run (execute) a Python program, we need to have a Python interpreter installed on our computer or we can use any online Python interpreter. The interpreter is also called Python shell. A sample screen of Python interpreter is shown in Figure 4.1.

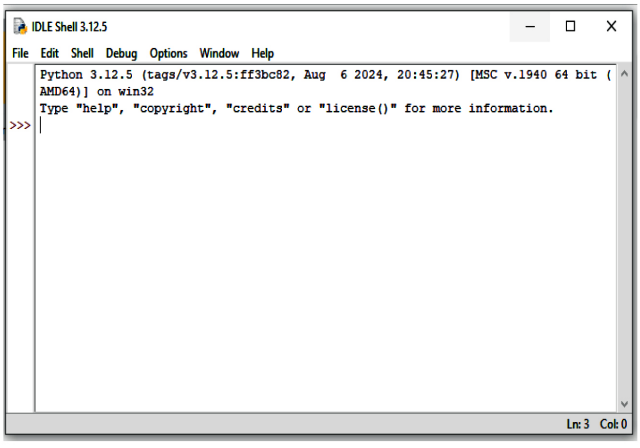


Figure 4.1

4.1 Revision of the Basics of Python

Input/Output statements (I/O):

I/O statements help us to interact with the program.

1. **Print Statement:**

   We use a '**print**' statement to display the output of data that we want to show.

   Example: **print(**"Hello, Python!"**)**

2. **Input Function:**

   The '**input**' function allows us to provide input to the program.

   Example: number = **input(**"Enter number: "**)**

Data Types and Variables

1. Data Types

Python also offers various data types which are listed as:

i. **Integer (int)**: It is a whole number ranging from negative infinity to positive infinity.

   Example:

   -,...,-3,-2,-1,0,1,2,3,....,

ii. **Float (float)**: It is numbers with decimals.

   Example:

   3.14, -0.5, 1.567.

iii. **String (str)**: It consists of alphabets, special characters, alphanumeric values which are enclosed in double quotes.

   Example:

   "hello" , "Python@", "Bhaktapur1", "@#@#kathmandu"

iv. **Boolean (bool):** It only provides **True** or **False** values.

Example:

is_student = **True**, has_mobile=**False**

v. **Identifier**

Identifiers are names given to program units such as variables, functions, classes, or other entities.

2. Variables

Variables are like containers that hold information. A variable is created when the value is assigned to it. **Example**: a=15

Operators and Expressions: Arithmetic, Relational, Logical, Assignment

Operators are special symbols that we use to do different things with numbers and words which allows us to perform specific actions.

1. Arithmetic Operator

Arithmetic operators are used in Python to do mathematical operations.

Operator	Name	Example
+	Addition	10 + 20 = 30
-	Subtraction	10 – 20 = -10
*	Multiplication	10 * 20 = 200
/	Division	20 / 10 = 2
%	Modulus	20 % 10 = 0
**	Exponent	10**20 =10**20
//	Floor Division	9//2 = 4

2. Relational Operator

Relational operators are used to check and compare values.

Name	Operator	Description	Example
Equal to	==	Two things are exactly the same	(5 == 5)
Not equal to	!=	Two things are not the same	(3 != 5)
Greater than (separate)	>	One is greater than the other	(7 > 5)
Less than	<	One is less than the other	(3 < 9)
Greater than or equal to	>=	Is greater or equal to another	(8 >= 8)
Less than or equal to	<=	Is less or equal to another	(4 <= 6)

3. Logical Operator

Logical operators in Python are used to combine conditions and make decisions based on different situations.

Operator	Name	Example
and	AND	a and b
or	OR	a or b
not	NOT	not(a)

Conditional Statement

The most fundamental form of conditional statement is **if** statement, "**if - else**" statement and "**if - elif - else**" statement.

if statement:

"if statement" is a conditional statement that gives us output based on the requirement of the condition that we provide.

Syntax

if condition**:**

#statement to be executed when the condition is true

Example:

#if statement program to check if a number is positive

number = **int(input**("Enter a number: "**))**

if number > 0:

print("The number is positive.")

Output:

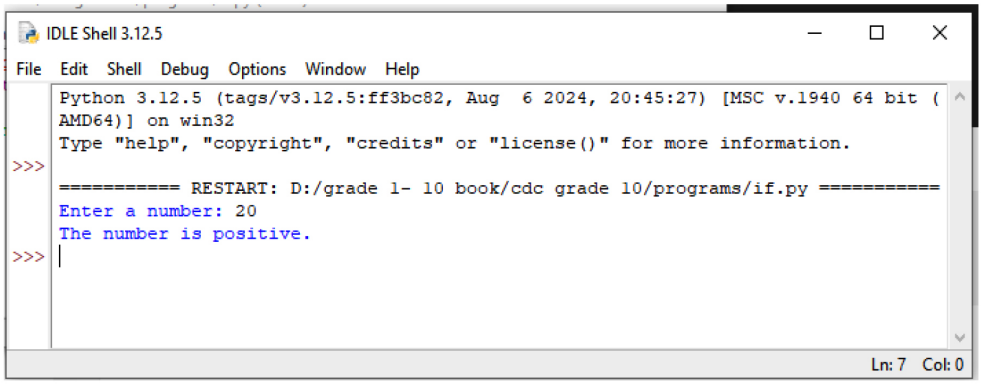


Figure 4.2

if-else condition

If else statements are used for conditional execution, allowing blocks of code to run based on a certain condition.

Syntax

if** condition**:

#statement to be executed when the condition is true

else:

#statement to be executed when the condition is false

Example:

```
# Asking age to the user and providing results.
user_age = int(input("How old are you? "))
if  user_age >= ١٨:
    print("You are an adult!")
else:
    print("You are a teenager or  a kid.")
```

Output

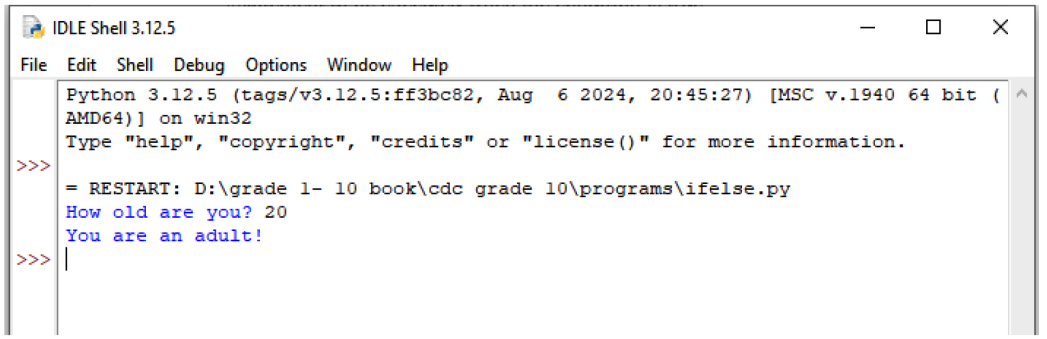


Figure 4.3

if-elif-else condition

Syntax

```
if condition1:
    #code to be executed if condition١ is True
elif condition2:
    #code to be executed if condition٢ is True
else:
    #code to be executed when condition١ and condition٢ are False
```

Example

```
# Checking the number's category, whether it is positive, negative or zero.
user_number = int(input("Enter a number: "))
if user_number > 0:
    print("The number is positive.")
elif user_number == 0:
    print("The number is zero.")
else:
    print("The number is negative.")
```

Output:

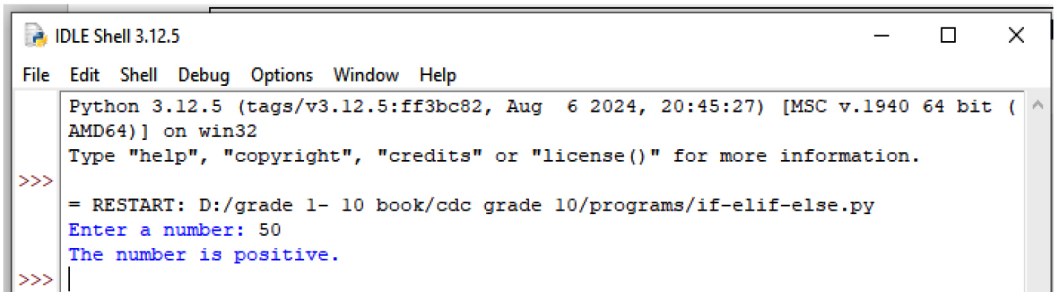


Figure 4.4

Nested if (if inside if)

Nested if statement is a construct where we put another if statement inside an existing if statement.

Syntax:

```
if condition1:
        #code to be executed if condition1 is True
    if condition:
            #code to be executed if condition2 is True
    else:
            #code to be executed when condition2 is False
else:
#code to be executed when condition1 and condition2 are False
```

Example:

```
age = int(input("Enter your age: "))
if age >= 16:
    print("You are eligible for citizenship.")
    if age >= 18:
        print("You are eligible to cast a vote.")
    else:
        print("You are not eligible to cast vote.")
else:
    print("You are a minor.")
```

Iteration

Iteration means doing things repeatedly. Iteration is the process of repeating a particular task until a specified condition is satisfied. The most fundamental example of iteration is the **"for"** loop and **"while"** loop.

for loop:

"For loop" is used when we know how many times, we want to repeat a block of code.

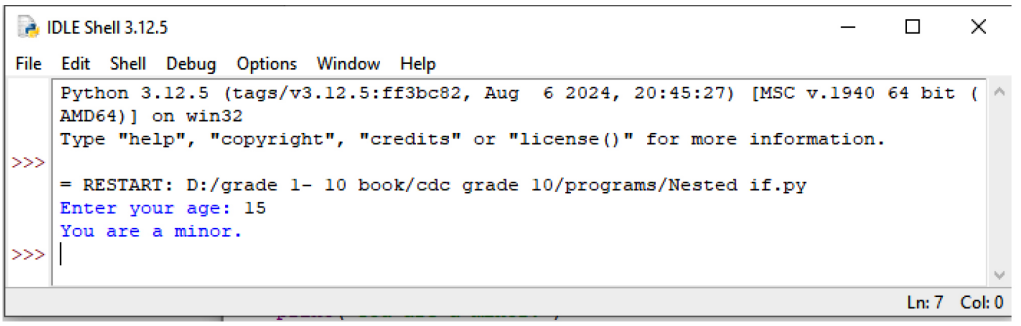


Figure 4.5

Syntax:

for *item* ***in*** *sequence:*

    # Code to do something with each item

Example:

#Using **for** loop to print **"Programming"** five times

```
for x in range(5):
 print("Programming")
```

Output

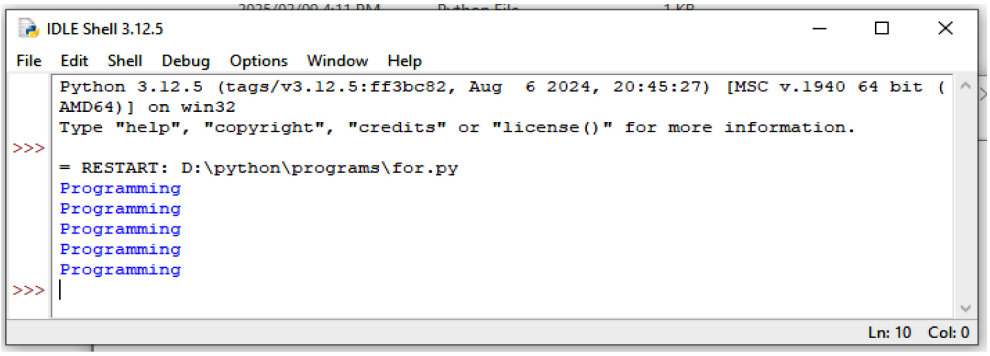


Figure 4.6

while loop:

"While loop" in Python is a control structure that allows us to repeatedly execute a block of code as long as certain conditions remain true.

Syntax

```
while condition:
```

Example:

```
# Using while loop to print numbers from 1 to 5
count = 1
while count <= 5:
    print(count)
    count += 1
```

Output

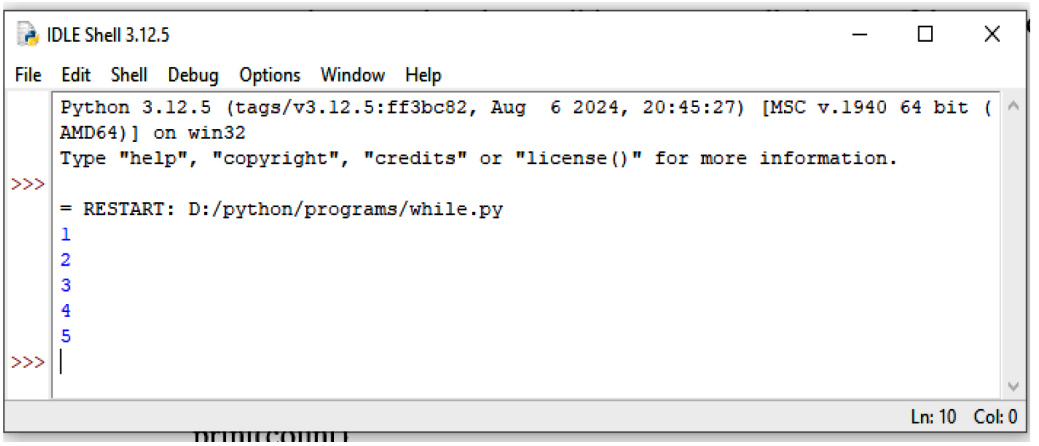


Figure 4.7

Python List

Python list is one of the built-in data types in Python used to store multiple data in a single variable.

Example:

thislist=["Computer", "Science", 20, True]

print(thislist)

Output:

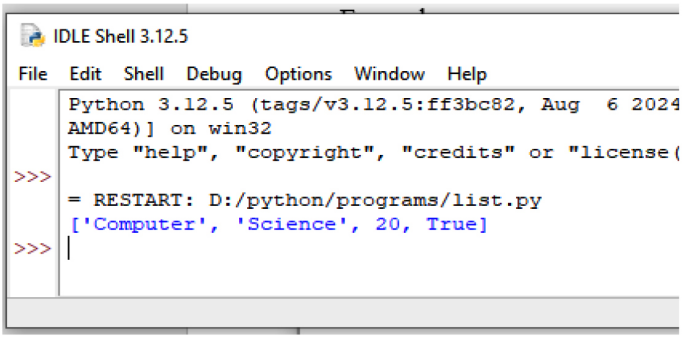


Figure 4.8

Activity 4.1

Write some program you have learnt in previous class.

Exercise

1. Choose the correct answer.

i. Which statement is used to display output in Python?
a) input ( ) b) print ( )
c) display ( ) d) output ( )

ii. Which of the following is NOT a valid Python data type mentioned in the text?
a) int b) float
c) character d) bool

iii. What type of values does the Boolean data type hold?
a) Whole numbers b) Decimal numbers
c) Text d) True or False

iv. Which of the following is an example of a relational operator?
a) + b) * c) == d) and

v. Which logical operator returns True if both conditions are true?
a) or b) not c) and d) !=

vi. What is the purpose of the if statement in Python?
a) To repeat a block of code.
b) To define a function.
c) To execute a block of code only if a condition is true.
d) To store multiple items in a single variable.

vii. Which conditional statement allows you to check multiple conditions in sequence?
a) if b) if-else
c) nested if d) if-elif-else

viii. What is the term for repeating a block of code multiple times?
a) Selection b) Iteration
c) Condition d) Assignment

ix. Which type of loop is used when you know the number of times you want to repeat a block of code?

a) while loop b) for loop

c) if loop d) nested loop

x. Which data structure is used to store multiple items in a single variable in Python, as shown in the example ["Computer", "Science", 20, True]?

a) String b) Tuple

c) List d) Dictionary

2. Write short answers to these questions.

a) What is the primary function of the input() function in Python?

b) Provide an example of a string literal in Python.

c) What is an identifier in the context of Python programming?

d) Explain the difference between the division operator (/) and the floor division operator (//) in Python.

e) Define the concept of data types in Python.

f) What is the purpose of relational operators in Python? Give one example.

g) Describe the functionality of the else block in an if-else statement.

h) What is a nested if statement?

i) When would you typically use a for loop instead of a while loop?

j) What is the role of the range() function often used with for loops?

k) Define a Python list and provide a simple example.

l) Discuss the concept of iteration in programming.

3. Write long answers to these questions.

a) Explain the different categories of operators available in Python (arithmetic, relational, and logical) with examples of each.

b) Describe the three main types of conditional statements in Python (if, if-else, and if-elif-else).

c) Explain the two types of loops available in Python (for and while) with their respective syntaxes and provide a practical example demonstrating the use of each loop.

d) Describe the four basic data types covered in the text (integer, float, string, and boolean), providing an example of each. Why is it important to understand data types when writing Python programs?

e) Imagine you need to write a Python program that takes an integer input from the user and performs the following actions:

   i. Prints "Positive" if the number is greater than 0.

   ii. Prints "Negative" if the number is less than 0.

   iii. Prints "Zero" if the number is equal to 0.

   iv. If the number is positive, it then checks if it's even or odd and prints the result.

f) Write the Python code for this program, demonstrating your understanding of input/output, conditional statements (including nested if), and basic arithmetic operators.

4.2 User-defined Functions

A Python function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

Types of Python Functions

1. **Built-in functions**

   Python's standard library includes number of built-in functions. Some of Python's built-in functions are print(), int(), len(), sum(), etc. These functions are always available, as they are loaded into computer's memory as soon as you start Python interpreter.

2. **User-defined functions**

   In addition to the built-in functions and functions in the built-in modules, you can also create your own functions. These functions are called user-defined functions.

User-defined functions

Functions that we define ourselves to do certain specific tasks are referred to as user-defined functions. The way in which we define and call functions in Python are already discussed in the previous section. Taking advantage of reusability feature of functions, there is many functions already available.

Creating User Defined Function

A function definition begins with def (short for define). The syntax for creating a user defined function is as follows:

def<Function name> ([parameter 1, parameter 2,....]):

set of instructions to be executed

[return <value>]

i. The items enclosed in "[]" are called parameters and they are optional. Hence, a function may or may not have parameters. Also, a function may or may not return a value.

ii. Function header always ends with colon (:).

iii. Function name should be unique. Rules for naming identifiers also applies for function naming.

iv. The statements outside the function indentation are not considered as part of the function.

Example

```
# Program to illustrate the use of user-defined functions
def add_numbers(x,y):
  sum = x + y
  return sum
num1 = 5
num2 = 6
print("The sum is", add_numbers(num1, num2))
```

Output:

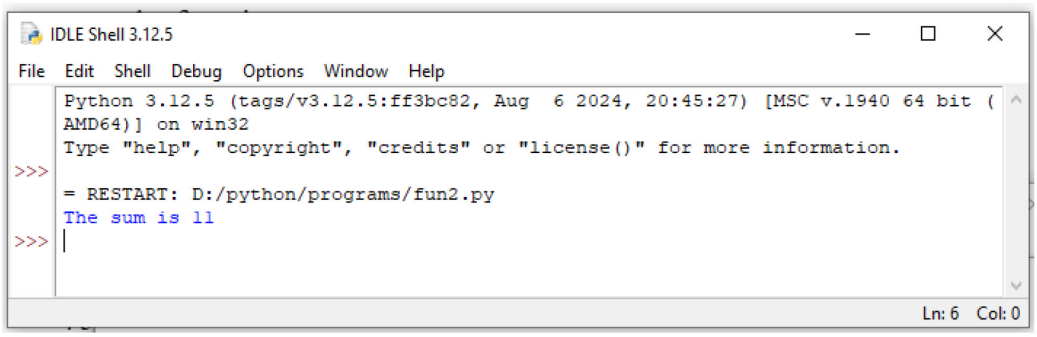


Figure 4.9

Scope of Function

The "scope of a user-defined function" refers to the specific part of a program where that function can be accessed and called, typically limited to the block of code where it is defined, meaning variables declared within the function are only accessible inside that function's body, not globally throughout the program; this is usually

called a "local scope" in programming languages. Generally, scope can be classified into following:

i. **Local scope:** Variables declared inside a function are only accessible within that function's code block.

ii. **Global scope:** If a variable is declared outside any function, it can be accessed from anywhere in the program.

Function Returns a Value

Till now we have discussed simple user-defined functions where no value was returned to the calling function or Python Interpreter. Now, we shall be discussing the functions with some return values. Usually, function definitions have the following basic structure:

Syntax:

def function name (arguments):

return <value>

Return command in Python specifies what value is to be returned back to the calling function. Function can take input values as parameters, execute them and return output (if required) to the calling function with a return statement.

Example

```
# Function to calculate the area using function
def area(length, width):
    return length * width
length = 5
width = 8
result = area (length, width)
print("Area of Rectangle:")
    print(result)
```

Output

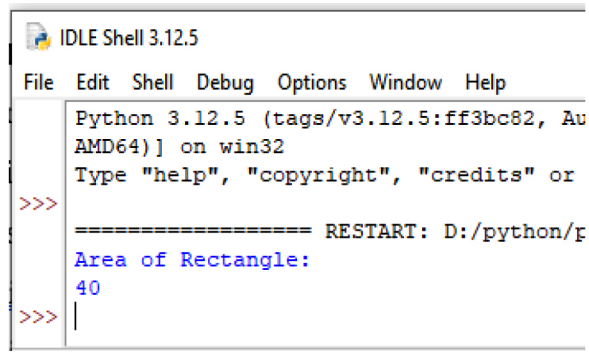


Figure 4.10

Parameters and Arguments in Functions

Parameters are the value(s) provided in the parentheses when we write function header. These are the values required by function to work.

Example

```
# Function to calculate the area of circle using function
  def area_of_circle (Radius):
    area = Radius** 2 * 22/7
    return area
  Radius = float (input ("Please enter the radius of the given circle: "))
print (" The area of the given circle is: ", area_of_circle (Radius))
```

Example:

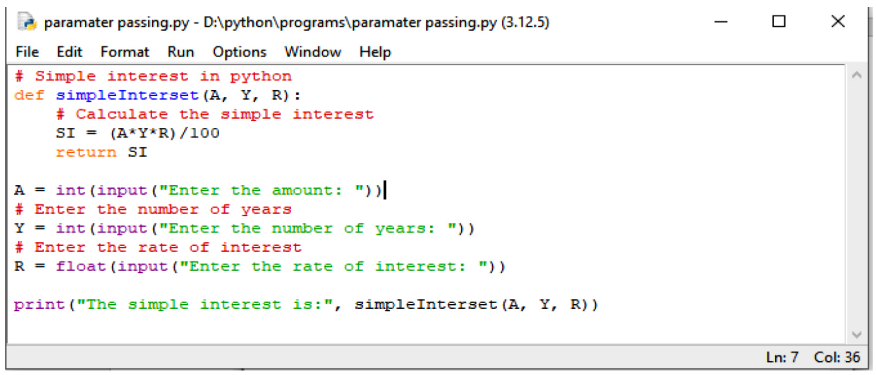


Figure 4.11

An argument is a value that is passed to the function when it is called. In other words, arguments are the value(s) provided in function call/invoke statement. List of arguments should be supplied in the same way as parameters are listed.

Example of argument in function call:

>>> area (5)

5 is an argument. An argument can be constant, variable, or expression.

Scope of Variable

In Python, user-defined functions can have different scopes, which determine where the function and its variables can be accessed. Here are some common scopes of user-defined functions:

1. Local Scope

Variables declared inside a function are only accessible within that function. When a variable is created inside the function, the variable becomes local to it. A local variable only exists while the function is executing

Example:

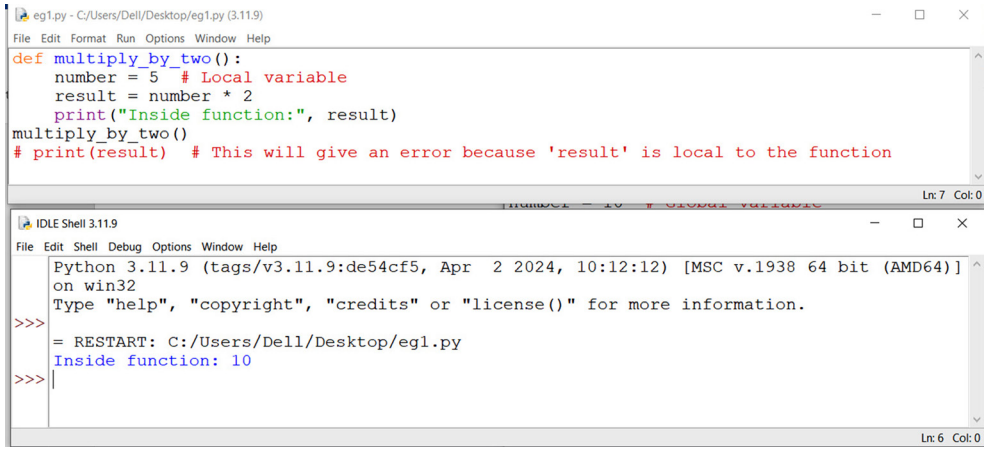


Figure 4.12

2. Global Scope

A function defined at the top level of a script or module can be accessed from anywhere in the module.

Example:

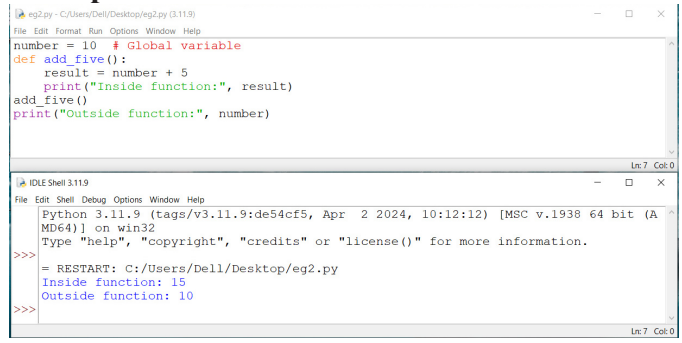


Figure 4.13

Passing Parameters

We have learnt that a function call must provide all the values as required function definition. If a function header has three parameters named in its header, then the function call should also pass three values. Python supports three types of formal arguments or parameters:

1. Positional arguments (Required arguments)

2. Default arguments

3. Keyword (or named) arguments

1. Positional/Required Arguments

When you create a function, you need to call statement for a given function definition. You need to match the number of arguments with number of parameters required.

For example, if a function definition header is like

```
def check (a, b, c):
```

then function calls for this can be:

```
check ( x, y, z)        # 3 values (all variables) passed
check (2, x, y)         #3 values (literal variables) passed
check (2, 5, 7)         #3 values (all literals) passed
```

2. Default Arguments

If we already know the value for a certain parameter, e.g., in an interest calculating function, we know that mostly the rate of interest is 10%, then there should be a provision to define this value as the default value.

Python allows us to assign default value(s) to a function's parameter(s) which is useful in case a matching argument is not passed in the function call statement. Default values are specified in the function header of function definition. Following is an example of function header with default values:

def interest (principal, time, rate 0.10):

here 0.10 is the default value for parameter rate

The default value is specified in a manner syntactically similar to a variable initialization.

Example

def Interest (prin, time, rate =0.10):	**#legal**
def Interest (prin, time=2, rate):	**#illegal**(default parameter before required parameter
def Interest (prin =2000, time=2, rate):	**#illegal**(default parameter before required parameter
def Interest (prin, time =2, rate= 0.10):	**#legal**
def Interest (prin =200, time = 2, rate= 0.10):	**#legal**

3. Keyword (Named) Arguments

The default arguments give you flexibility to specify the default value for a parameter so that it can be skipped in the junction call, if needed. However, still you cannot change the order of the arguments in the function call; you must remember the correct order of the arguments.

Python offers another type of argument: **keyword arguments**.

Python offers a way of writing function calls where you can write any argument in any order provided you name the arguments when calling the function, as shown below:

interest (prin 2000, time 2, rate =0. 10)

interest (time prin = 2600, rate= 0.09)

interest (time = 2, rate = 0.12, prin =2000)

All the above function calls are valid now, even if the order of arguments does not match the order of parameters as defined in the function header.

Returning values from functions

Functions in Python may or may not return a value. You already know about it. There are two types of functions in python:

1. Functions returning value (non-void function)

2. Functions not returning any value (void functions)

l. Functions returning some value (non-void functions)

The functions that return some computed result in terms of a value, fall in this category. The computed value is returned using return statement as per syntax

return <value>

The value being returned can be one of the following:

a. literal

b. a variable

c. an expression

Example

def sum (x., y):

return s

We are invoking this function as:

result = sum (5, 3)

Here, the return value from sum () will replace function call.

2. Functions not returning any value (void functions)

The functions that perform some action or do some work but do not return any computed value or final value to the caller are called void functions. A void function may or may not a return statement. If a void function has a return statement, then, it takes the following form:

return

That is, keyword **return** without any value or expression. Following are some examples of void function:

```
def greet ( ):
        print("Hello")

def quote ( ):
   print ("python is good")
return
```

```
def greet1 ( ):
        print ("Hello", name)

def printsum (a, b, c):
    print ("Sum is, a+ b + c")
return
```

The void functions do not return a value, but, they **return** a legal **empty** value of python.

Example

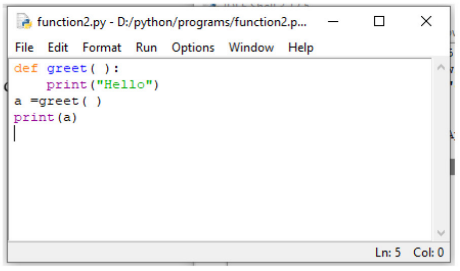


Figure 4.14

Output

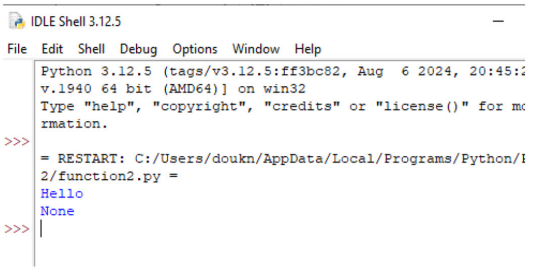


Figure 4.15

Returning Multiple Values

To return multiple values from a function, you have to ensure the return statement inside a function body should be of the form given:

return <valuel/ variablel/ expressionl>, <value2/variable2/expression2>,

Example

```
def squared(x, y, z):
    return x* x, y*y, z * z
t=squared(2,5,7)
    print(t)
```

Output

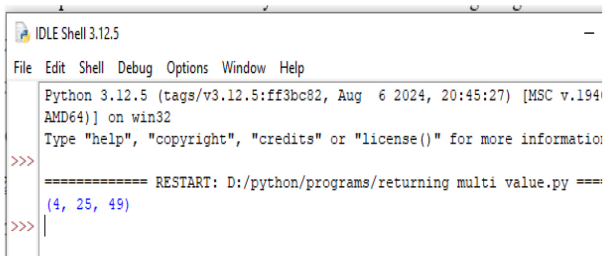


Figure 4.16

Example

```
def squared(x, y, z):
 return x* x, y*y, z * z
 v1, v2, v3 = squared(2,3,4)
 print("the returned values are
 as under: " )
 print(v1 ,v2, v3)
```

Output

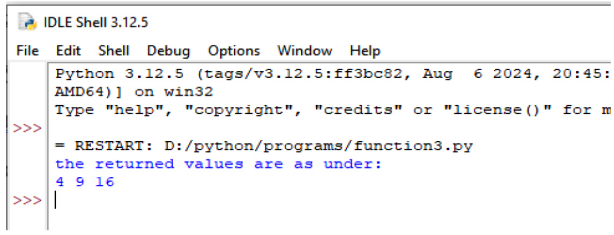


Figure 4.17

Exercise

1. Choose the correct answer.

i. Which keyword is used to define a user-defined function in Python?

a) function b) define
c) def d) func

ii. What are the values passed to a function when it is called known as?

a) Parameters b) Arguments
c) Return values d) Scope

iii. The part of the program where a function can be accessed is called its:

a) Parameter b) Argument
c) Return type d) Scope

iv. **Variables declared inside a function have:**

a) Global scope b) Local scope

c) Unlimited scope d) No scope

v. **What does the return statement do in a Python function?**

a) Prints output to the console.

b) Takes input from the user.

c) Sends a value back to the caller of the function.

d) Defines the function.

vi. **What are the values specified in the function header within the parentheses called?**

a) Arguments b) Parameters

c) Return values d) Local variables

vii. **What type of argument allows you to call a function by specifying the parameter names?**

a) Positional arguments b) Default arguments

c) Keyword arguments d) Required arguments

viii. **A function that performs an action but does not explicitly return a value is sometimes referred to as a:**

a) Non-void function b) Void function

c) Recursive function d) Anonymous function

ix. **Can a Python function return multiple values?**

a) No b) Yes, as a list

c) Yes, as a tuple d) Yes, directly separated by commas

2. Write short answers to these questions.

a) What is the main benefit of using user-defined functions in programming?

b) Name the two types of Python functions.

c) Explain the difference between a parameter and an argument in the context of Python functions.

d) What is local scope in Python functions? Provide a brief example.

e) What is global scope in Python? How does it differ from local scope?

f) What is the purpose of the return keyword in a Python function?

g) Explain what positional arguments are and how they are passed to a function.

h) Describe the use case for default arguments in Python functions.

i) What are keyword arguments, and what advantage do they offer when calling a function?

j) What happens if a void function has a return statement without any value?

3. Write long answer to these questions.

a) Explain the concept of function scope in Python, differentiating between local and global scope.

b) Describe the concept of return values in Python functions. Differentiate between functions that return a value (non-void) and those that do not (void). Provide examples of both types and explain how the return value can be used in the calling part of the program.

c) Python allows functions to return multiple values. Explain how this is achieved and provide an example demonstrating a function that returns multiple values and how these values can be accessed by the caller.

d) Design a Python program that includes at least two user-defined functions:

   i. One function that takes two numbers as arguments and returns their product.

   ii. Another function that takes a list of numbers as an argument and prints each number.

4.3 Concept of Library and Packages in Python

In programming, if we create smaller handleable units, called as modules. A related term is library. A library refers to a collection of modules that together supply to specific type of needs or applications e.g., **NumPy** library of Python provides to scientific computing needs. Here, we shall talk about using some Python libraries.

Python Library

A library is a collection of modules (and packages) that together provides a specific type of applications or requirements. A library can have multiple modules in it.

Commonly used Python libraries are as listed below:

i. **NumPy**: A big library for science, it supports big arrays and lots of math functions.

ii. **Pandas**: It helps with data, offering tools for working with different kinds of data.

iii. **Matplotlib**: It's for making charts, static or interactive, and is often used with Pandas.

Python Package

A package is a container that contains various functions to perform specific tasks. Math package includes the sqrt ( ) function to perform the square root of a number.

Packages help in ensuring the reusability of code. To access any module or file from a Python package, use an import statement in the Python source file where you want to access it. Import of modules from Python Packages is done using the dot operator (.).

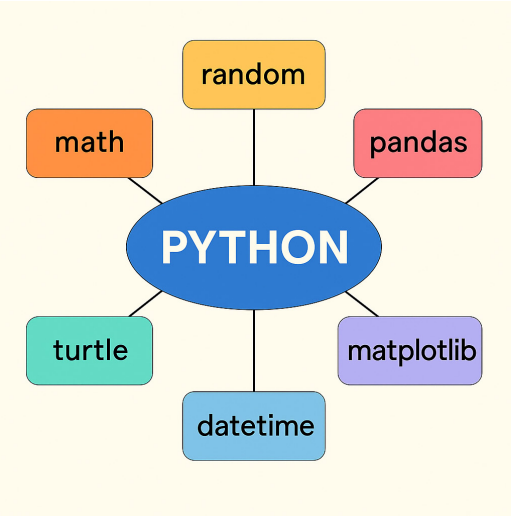


Figure 4.18

Syntax:

import module1[, module2,... moduleN]

4.3.1. Importing and use of standard libraries

Python libraries are collections of code and information that give the Python programming language center functionality. Python comes with a set of standard libraries that provide built-in modules to perform common tasks such as file handling, mathematics, system operations, and more. These libraries can be imported using the import statement.

Syntax

library_name, like so:

Example

import requests

Python provides different ways to import and use standard libraries:

a. Import the entire module

```
import math
print(math.sqrt(25))
```

Output: 5.0

You need to use module_name.function_name() to access functions.

b. Import a specific function

```
from math import sqrt
print(sqrt(25))
```

Output: 5.0

No need to prefix with math. when calling sqrt().

c. Import a module with an Alias (Shortcut)

```
import datetime as dt

print(dt.datetime.now())  # Prints current date and time as dt is used as a shorthand for datetime.
```

d. Import all functions from a module

```
from math import *

print(sin(90))  # Output depends on radians
```

Output : 0.8939966636005579

4.3.2 Introduction to popular libraries (Math, Random, Pandas, Turtle and Matplotlib)

As you have studied earlier, libraries can be imported using the import statement.

Syntax

library_name, like so:

Libraries are planned to be utilized by engineers to rapidly get to common usefulness, unique complex operations, or effortlessly reuse code. Some of the popular libraries used in python are:

1. Math

Math is a built-in library in Python that provides access to mathematical functions. It includes functions for basic math operations, trigonometry, logarithms, and more. It is generally used for mathematical operations that are not covered by NumPy or SciPy. Different mathematical functions are:

i. Functions for mathematical constants (e.g., math.pi).

ii. Trigonometric functions like sin(), cos(), and tan().

iii. Logarithmic functions (log(), log10()).

iv. Power functions like pow().

syntax.

import math

Example

```
import math
# compute the square root of 256
x = math.sqrt(256)
#assert x == 16
print("Squre root is  ", math.sqrt(16))
print("value is ", math.sin(math.pi/2))
print("Code works just fine, x is equal to ", x)
```

Output

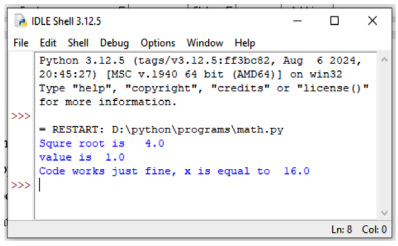


Figure 4.19

2. Random

Random function is used to generate random numbers and provide random selection. It is generally used for random sampling, generating random numbers for games, simulations, or security purposes. There are different random functions in the Random Module of Python. Some of them are:

i. randrange() Returns a random number within the range

ii. randint() Returns a random integer within the range

iii. choice() Returns a random item from a list, tuple, or string

iv. random() Generate random floating numbers

Example

```
import random

print(random.randint(1, 10))

print(random.choice(['apple', 'banana', 'Orange']))

list1 = [1, 2, 3, 4, 5, 6]

print(random.choice(list1))
```

```
import random

r1 = random.randint(5, 15)

print("Random number between 5 and 15 is % s" % (r1))

r2 = random.randint(-10, -2)

print("Random number between -10 and -2 is % d" % (r2))
```

Output:

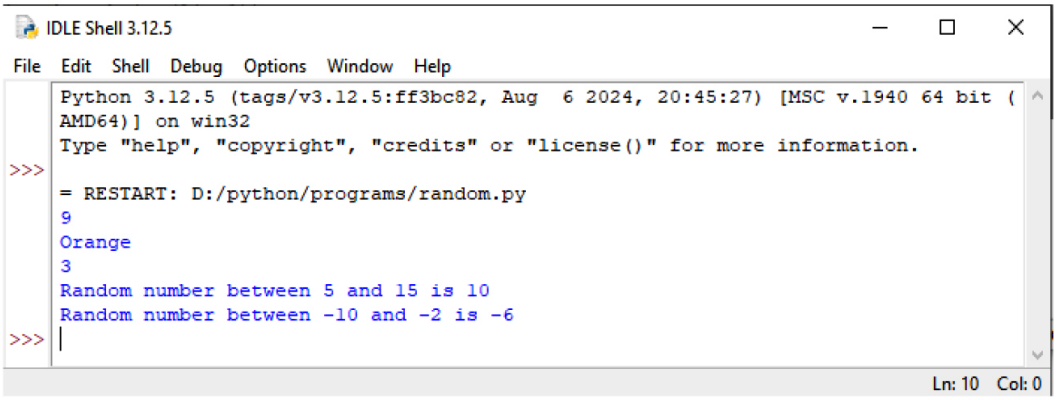


Figure 4.20

3. Pandas

Pandas offers necessary features for effortlessly manipulating, analyzing, and managing large datasets. To achieve this, Pandas utilizes methods like data frame operations, time series analysis, and data cleansing. It is powerful data structures for manipulating numerical tables and time series. It is also a comprehensive tool for reading and writing data between different formats. It supports reading and writing data to / from various file formats (CSV, Excel, SQL, etc.). To run panda, we need to install it using following command.

$ pip install pandas

Some of the most important Pandas are

pandas.read_csv()	Loads data from a CSV file into a table-like structure called a DataFrame.
DataFrame.info()	Shows a summary of the DataFrame: column names, data types, and missing values.
DataFrame.shape	Returns the size of the DataFrame as a tuple (rows, columns).
pandas.DataFrame()	Creates a table (DataFrame) with rows and columns that can hold different types of data

Example

```
import pandas as pd
data = {'Name': ['Shyam', 'Sanskar'], 'Age': [25, 30]}
df = pd.DataFrame(data)
print(df)
```

Output:

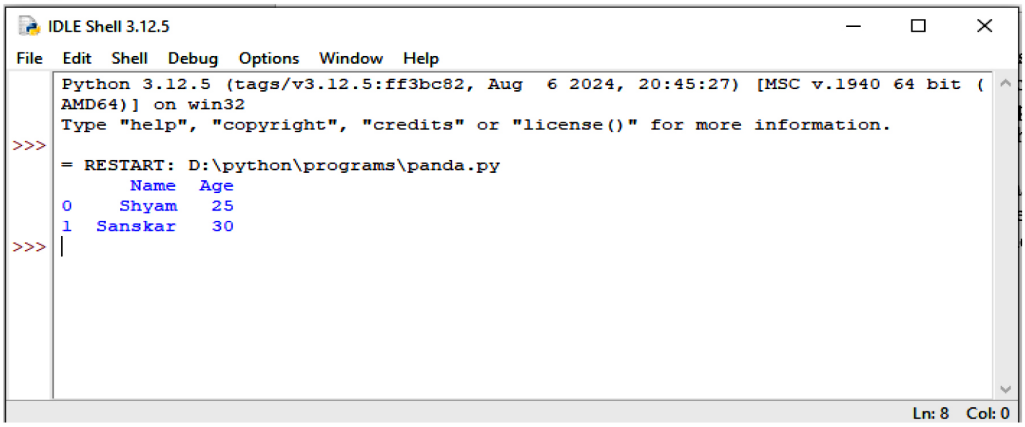


Figure 4.21

If the pandas library is not installed on your system, you will encounter an error. You can solve this by installing the library using pip. Here is the process for it:

1. Open your command prompt or terminal.
2. Run the following command to install pandas:

   pip install pandas

4. Turtle

The turtle module in Python is a pre-built module in Python that can be used to create a graphical representation on the terminal using a cursor which is known as a turtle. Turtle is a module in Python that allows us to bring **shapes**, **figures**, and **designs** to life on a screen. It is a simple interface for drawing shapes, lines, and animations on a screen. It is used for teaching programming through fun and interactive graphics. The list of some of the most important functions in turtle are:

i. **forward():** Moves the turtle forward by a specified amount

ii. **backward():** Moves the turtle backward by a specified amount

iii. **right():** Turns the turtle clockwise by a specified angle

iv. **left():** Turns the turtle counterclockwise by a specified angle

v. **goto():** Moves the turtle to a specified position

vi. **pendown():** Puts down the turtle's tail so that it draws when it moves

vii. **penup():** Stops the turtle from drawing

viii. **Turtle():** Creates and returns a new turtle object

ix. **mainloop():** Tells the window to wait for the user to do something

Syntax

import turtle

Example

```
import turtle
s = turtle.Turtle()
for i in range(4):
    s.forward(50)
    s.right(90)
turtle.done()
```

Output:

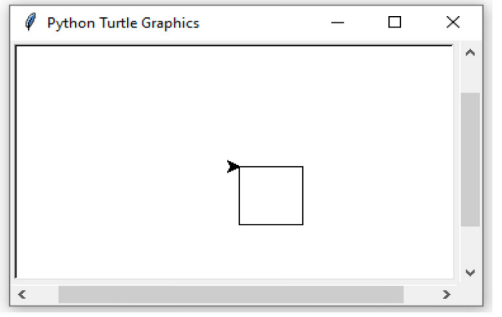


Figure 4.22

5. matplotlib

Matplotlib is a low-level graph plotting library in python that serves as a visualization utility for high-quality visualizations and graphs. It was created by John D. Hunter. Matplotlib is open source and we can use it freely. It can create different types of visualization reports like line plots, scatter plots, histograms, bar charts, pie charts, box plots, and many more different plots. This library also supports 3-dimensional plotting.

Before using we should install Matplotlib by using the **PIP** command.

pip install matplotlib

There are different types of plots in Matplotlib. Some of them are:

i. Line plot: A line plot shows the relationship between the x and y-axis.

ii. Bar Plot: the barplot shows the relationship between the numeric and categoric values

iii. Pie charts: A pie chart (or circular chart) is used to show the percentage of the whole.

Example

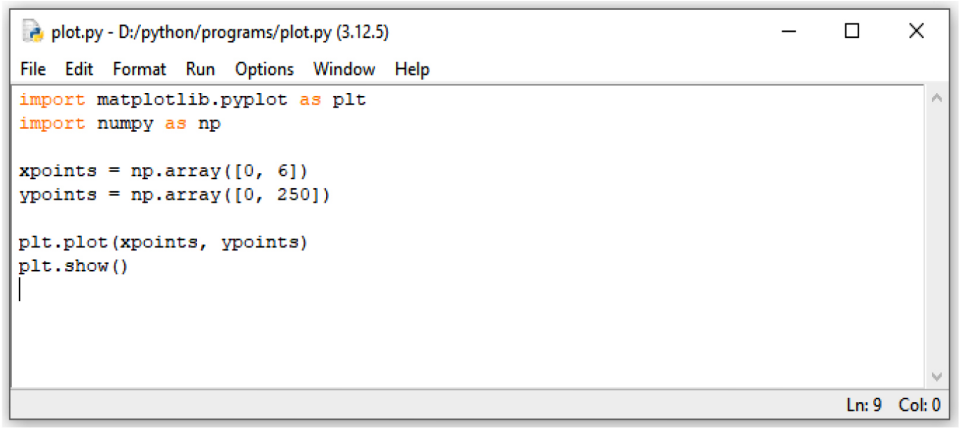


Figure 4.23

Output:

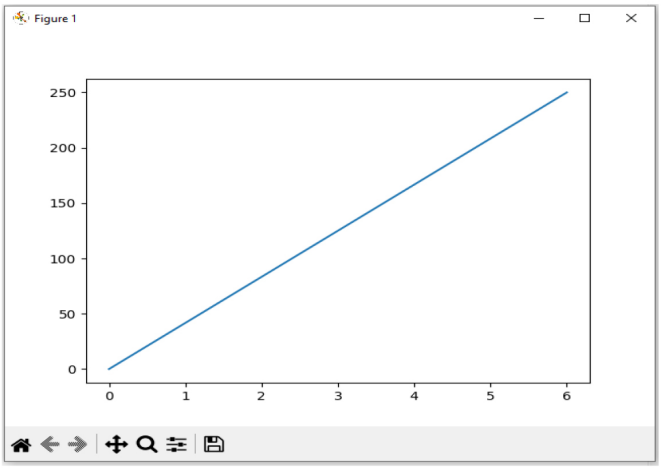


Figure 4.24

4.4 Graphics Using Turtle

Define, draw turtle, function (Forward, Backward, Left, Right, Penup, color, Fillcolor), shape.

The turtle module in Python is a pre-built module in Python that can be used to create a graphical representation on the terminal using a cursor which is known as a turtle. Turtle is a module in Python that allows us to bring **shapes**, **figures**, and **designs** to life on a screen. The list of some of the most important functions in turtle are:

Method	Parameter	Description
forward()	amount	Moves the turtle forward by the specified amount
backward()	amount	Moves the turtle backward by the specified amount
right()	angle	Turns the turtle clockwise
left()	angle	Turns the turtle counterclockwise
penup()	None	Picks up the turtle's Pen
color()	Color name	Changes the color of the turtle's pen
fillcolor()	Color name	Changes the color of the turtle will use to fill a polygon

Use of Turtle Module

i. **Easy visualization of programming concepts:** Loops, functions, and variables are interesting to learn, and this page gives a basic and fun approach to learning such concepts.

ii. **Interactive learning:** Through the app, users can command the turtle in Python as to the directions it has to go and the shapes that it needs to draw.

iii. **Enhances creativity:** Absolute novices can make quite compelling patterns and designs with quite simple code in Python.

iv. **Simplified debugging:** Python is a high-level language and thus it is easier to debug and understand than languages such as C-graphics and so on.

Turtle Motion

The turtle can move forward and backward in direction that it's facing. Let's see the following functions.

a) **forward(*distance*) or turtle.fd(*distance*) -** It moves the turtle in the forward direction by a certain distance. It takes one parameter **distance,** which can be an integer or float.

Example

```
import turtle
sk = turtle.Turtle()
sk.forward(50)
turtle.done()
```

Output

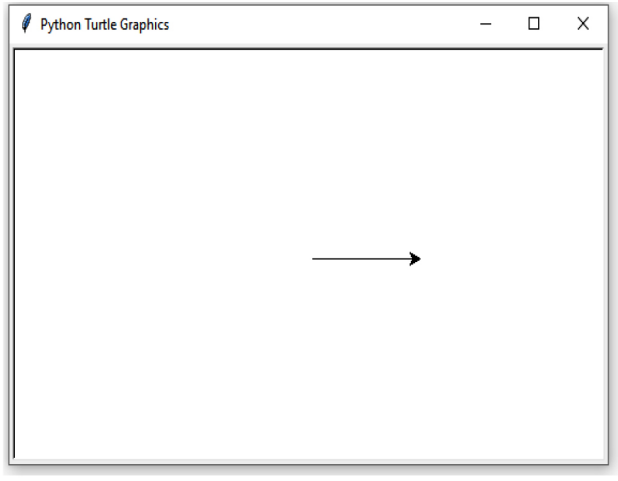


Figure 4.25

b) **back(distance)** or **turtle.bk** or **turtle.backward(distance)** - This method moves the turtle in the opposite direction the turtle is headed. It doesn›t change the turtle heading.

Example:

```
import turtle
sk = turtle.Turtle()
sk.backward(50)
turtle.mainloop()
```

Output:

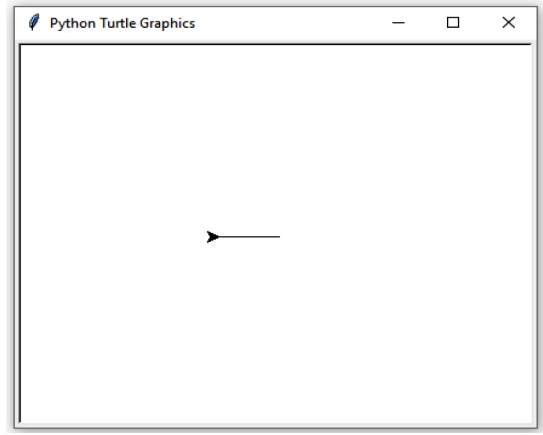


Figure 4.26

c) right(angle) or turtle.rt(angle) - This method moves the turtle right by angle units.

Example:

```
import turtle
t = turtle.Turtle()
t.heading()
t.right(30)
 t.heading()
turtle.mainloop()
```

Output:

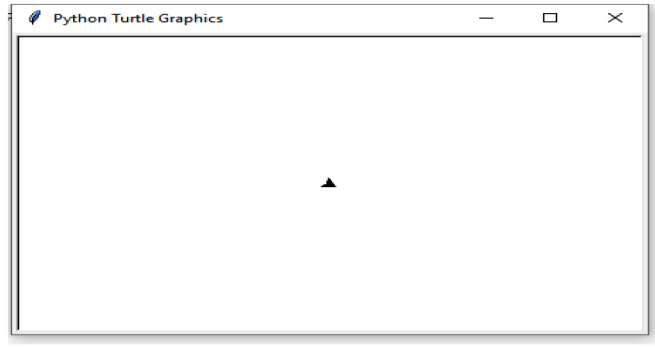


Figure 4.27

left(angle) or turtle.lt(angle) –

This method turns the turtle left by angle units.

Example:

```
import turtle
t = turtle.Turtle()
t.heading()
t.left(100)
t.heading()
turtle.mainloop()
```

Output:

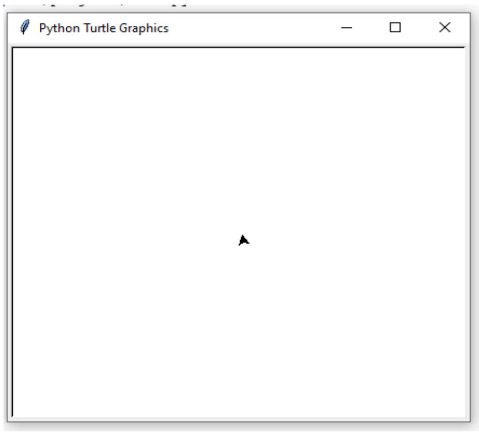


Figure 4.28

penup()

This will lift the turtle off the "digital canvas" and if you move the turtle in penup state it won't draw.

Example

```
import turtle
turtle.color("red")
turtle.hideturtle()
turtle.speed(1)
turtle.left(90)
for i in range(4):
    turtle.forward(30)
    turtle.penup()
    turtle.forward(30)
    turtle.pendown()
turtle.exitonclick()
```

Output:

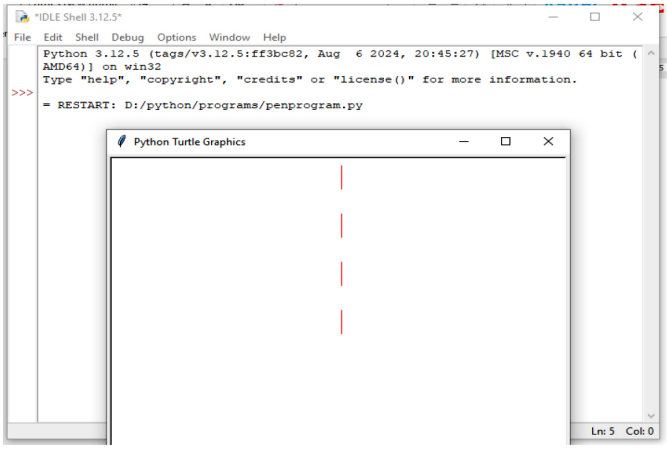


Figure 4.29

color() :

This method is used to change the color of the ink of the turtle drawing. The default color is black.

Syntax:

turtle.color(*args)

Example

```
import turtle
 turtle.forward(50)
 # change the color of turtle
turtle.color("blue")
turtle.forward(150)
turtle.color("red")
turtle.forward(50)
```

Output:

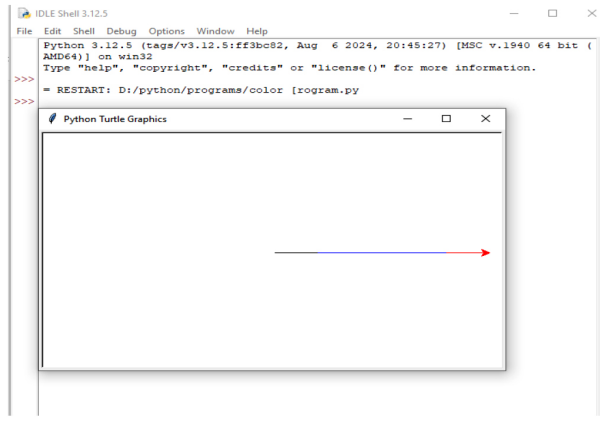


Figure 4.30

fillcolor(): This helps to choose the color for filling the shape. It takes the input parameter as the color name or hex value of the color and fills the upcoming closed geographical objects with the chosen color. We need following to implement it

begin_fill(): This function tells turtle that all upcoming closed graphical objects needed to be filled by the chosen color.

end_fill(): This function tells turtle to stop the filling upcoming closed graphical objects.

Example:

```
import turtle
 # creating turtle pen
t = turtle.Turtle()
```

```
# taking input for the radius of the circle
r = int(input("Enter the radius of the circle: "))
 # taking the input for the color
col = input("Enter the color name or hex value of color(# RRGGBB): ")
# set the fillcolor
t.fillcolor(col)
# start the filling color
t.begin_fill()
# drawing the circle of radius r
t.circle(r)
# ending the filling of the color
t.end_fill()
```

Output:

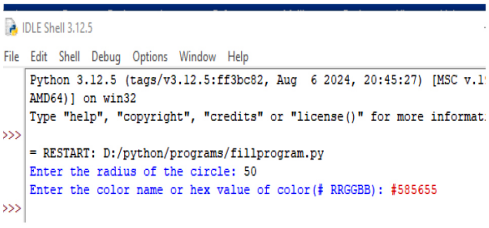


Figure 4.31

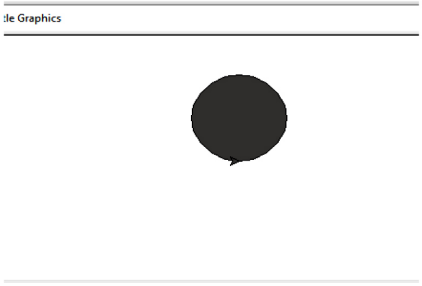


Figure 4.32

shape()

This function is used to set the turtle shape to shape with a given name or, if the name is not given, return the name of the current shape.

Syntax:

turtle.shape(name=None)

Polygon shapes available in Python are: “arrow”, “turtle”, “circle”, “square”, “triangle”, “classic”.

Example:

```
import turtle
# for default shape
turtle.forward(100)
# for circle shape
turtle.shape("circle")
turtle.right(60)
turtle.forward(100)
# for triangle shape
turtle.shape("triangle")
turtle.right(60)
turtle.forward(100)
# for square shape
turtle.shape("square")
turtle.right(60)
turtle.forward(100)
# for arrow shape
turtle.shape("arrow")
turtle.right(60)
turtle.forward(100)
# for turtle shape
turtle.shape("turtle")
turtle.right(60)
turtle.forward(100)
```

Output:

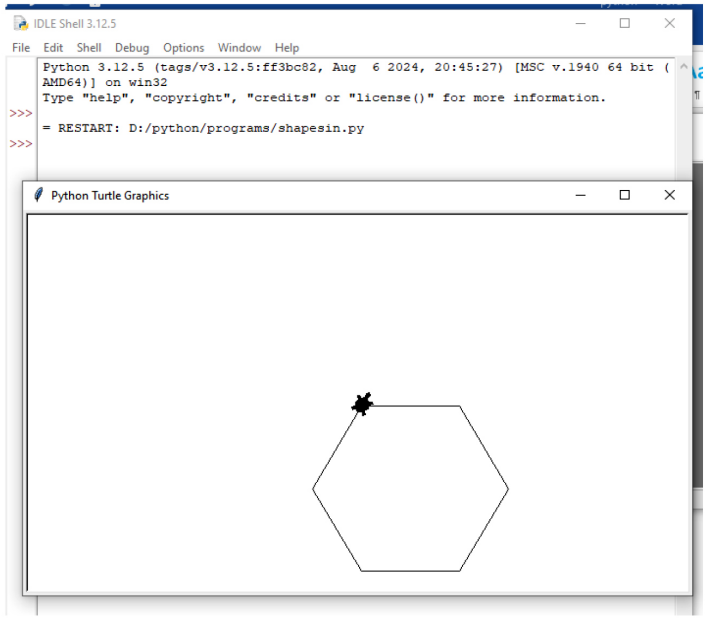


Figure 4.33

4.5 Error handling: errors and exceptions, try-except blocks

As a programmer, you know that errors are an inevitable part of the coding process. No matter how well you plan and code, things can (and often do) go wrong. That's where error handling comes into play. Error handling is a critical aspect of Python programming that empowers developers to identify, manage, and resolve errors, ensuring robust and error-free code. It plays a pivotal role in enhancing the reliability and maintainability of Python code.

Errors and Exceptions

Errors and exceptions can lead to unexpected behavior or even stop a program from executing. Python provides various functions and mechanisms to handle these issues and improve the robustness of the code.

1. Errors: An error is an issue in a program that prevents the program from completing its task. Problems in code that can cause it to fail. Types of error are:

i. Syntax Errors: Syntax errors occur when the code violates the rules of the programming language's syntax. Mistakes in the use of Python syntax (e.g., missing colons).

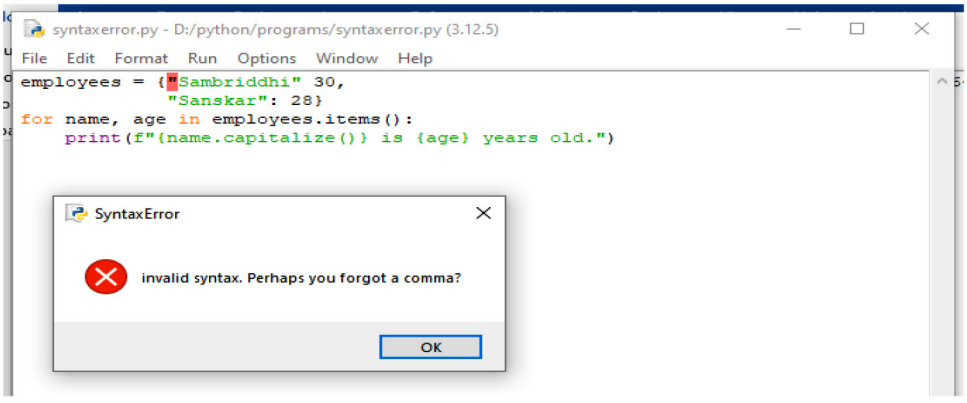


Figure 4.34

ii. Runtime Errors: A runtime error in a program is one that occurs after the program has been successfully compiled. Errors that occur during program execution division by zero.

Example:

```
numerator = int(input("Enter numerator: "))

denominator = int(input("Enter denominator: "))

result = numerator / denominator

print(f"Result: {result}")
```

Output

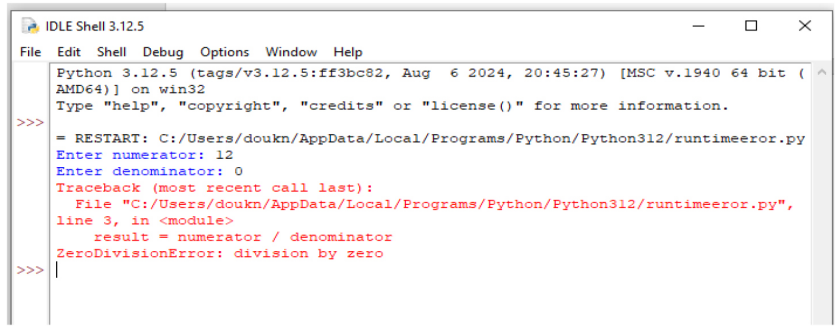


Figure 4.35

iii. **Logical Errors**: Logical errors in Python occur when the program runs without raising any syntax or runtime errors, but the output is incorrect or doesn't match the expected result. Bugs in the code that result in incorrect output (e.g., wrong algorithm implementation).

Example:

```
Firstnum = int(input("Enter Firstnum: "))

secondnum = int(input("Enter secondnum: "))

result = (Firstnum + secondnum)/2

print("The Result is : {result}")
```

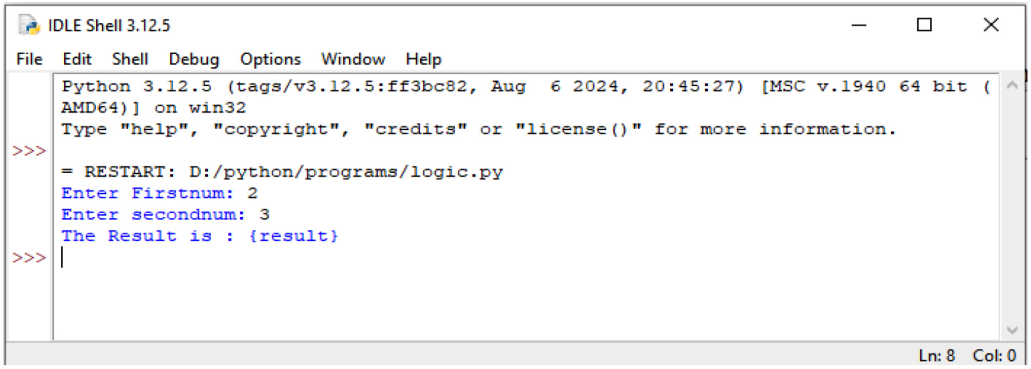


Figure 4.36

iv. Exceptions: Even if a statement or expression is syntactically correct, it may cause an error when an attempt is made to execute it. Errors detected during execution are called ***exceptions*** and are not unconditionally fatal. Most exceptions are not handled by programs. In specific types of runtime errors, Python can handle using its built-in mechanisms. When an exception occurs, Python generates an error message and halts execution unless the exception is handled. It is possible to write programs that handle selected exceptions.

Example:

divide_by_zero = 7 / 0

The above statement causes an exception as it is not possible to divide a number by **0**.

2. Exception Handling

After learning about errors and exceptions, we will learn to handle them by using try, except, else, and finally blocks.

So, what do we mean by handling them? In normal circumstances, these errors will stop the code execution and display the error message. To create stable systems, we need to anticipate these errors and come up with alternative solutions or warning messages.

try, except, else and finally Blocks

i. **try Block**: try block lets us test a block of code for errors. Python will "try" to execute the code in this block. If an exception occurs, execution will immediately jump to the except block.

ii. **except Block:** except block enables us to handle the error or exception. If the code inside the try block throws an error, Python jumps to the except block and executes it. We can handle specific exceptions or use a general except to catch all exceptions.

iii. **else Block:** else block is optional and if included, must follow all except blocks. The else block runs only if no exceptions are raised in the try block. This is useful for code that should execute if the try block succeeds.

iv. **finally Block:** finally block always runs, regardless of whether an exception occurred or not. It is typically used for cleanup operations (closing files, releasing resources).

Example:

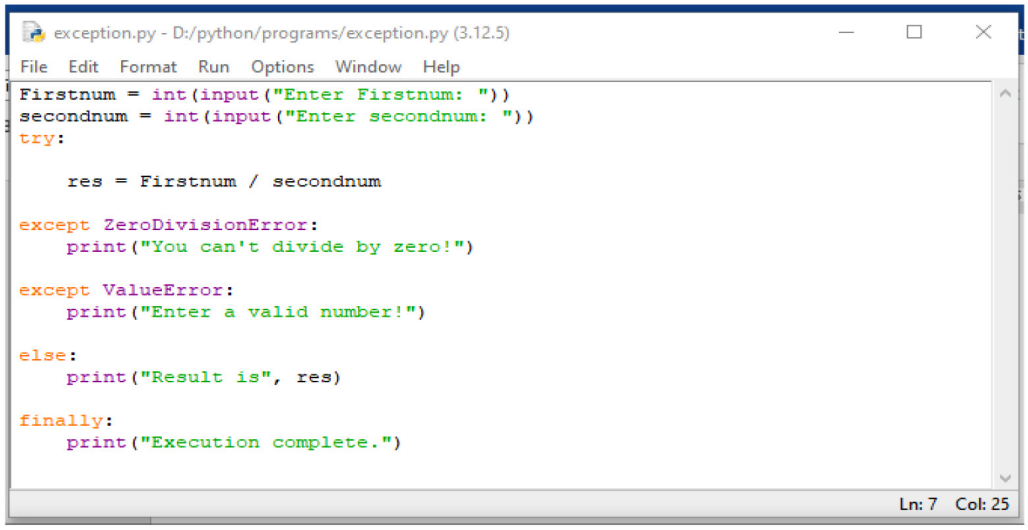


Figure 4.37

Output

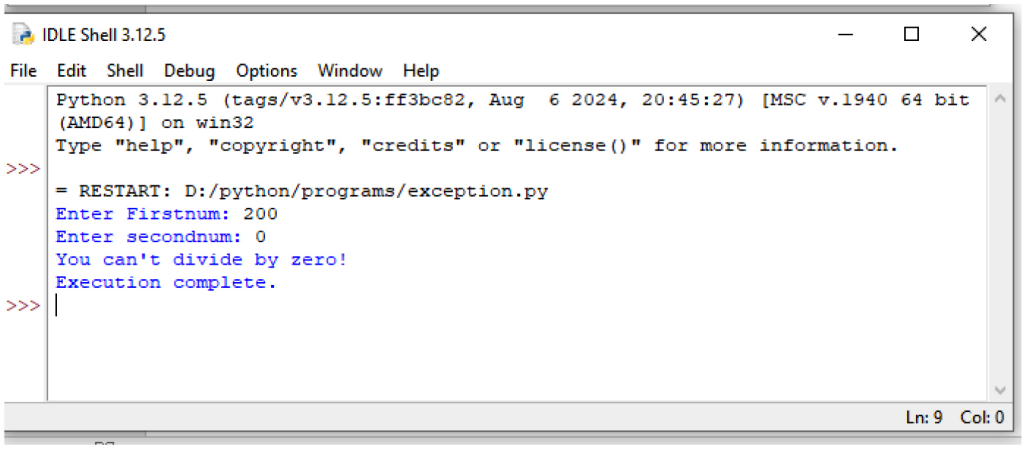


Figure 4.38

Exercise

1. Choose the correct answer.

i. A collection of modules that together supply to specific needs or applications is called a:

a) Module b) Package
c) Library d) Function

ii. Which of the following is a popular Python library used for data manipulation and analysis?

a) Math b) Random
c) Pandas d) Turtle

iii. What is a container that contains various functions to perform specific tasks in Python?

a) Module b) Library
c) Package d) Function

iv. **Which statement is used to access modules or files from a Python package?**

a) include b) use
c) import d) access

v. **Which of the following is NOT a standard way to import a module in Python?**

a) import module_name
b) from module_name get function_name
c) import module_name as alias
d) get function_name from module_name

vi. **Which Python library provides functions for generating random numbers?**

a) Math b) Random
c) Pandas d) Matplotlib

vii. **What type of error occurs when the code violates the rules of the programming language's syntax?**

a) Runtime Error b) Logical Error
c) Syntax Error d) Exception

viii. **Errors detected during program execution are called:**

a) Syntax Errors b) Logical Errors
c) Exceptions d) Warnings

ix. **Which block is used to test a block of code for potential errors in Python?**

a) catch b) handle
c) try d) error

x. **Which block in error handling always executes, regardless of whether an exception occurred or not?**

a) else b) finally
c) except d) try

2. Write short answers to these questions.

a) Define the term "module" in the context of Python programming.

b) What is the relationship between modules and libraries in Python?

c) Give an example of a built-in Python library and its purpose.

d) Explain why packages are useful for code organization and reusability.

e) What is the purpose of using an alias when importing a module in Python? Provide an example.

f) Name two functions provided by the Python math library.

g) What is a runtime error? Give an example of a situation that might cause one.

h) Explain the difference between an error and an exception in Python.

i) What is the role of the except block in a try-except statement?

j) When is the else block in a try-except-else statement executed?

3. Write long answers to these questions.

a) Explain the concepts of Python libraries and packages, highlighting their importance in software development.

b) Introduce three popular Python libraries mentioned in the text: Pandas, Turtle, and Matplotlib. For each library, briefly describe its primary purpose and provide a simple example of how it can be used in a Python program.

c) Discuss the different types of errors that can occur in Python programs: syntax errors, runtime errors, and logical errors. Provide an example of each type and explain how they can impact the execution and correctness of a program.

d) Explain the importance of error handling in Python.

Sample Programs:

1. **Write a program to take two numbers as input and print their sum.**

```
x = int(input("Enter first number: "))
y = int(input("Enter second number: "))
print("Sum of", x, "and", y, "is", x + y)
```

2. **Write a program to check if a number is positive, negative, or zero.**

```
num = int(input("Enter a number: "))
if num > 0:
    print("The number is positive.")
elif num == 0:
    print("The number is zero.")
else:
    print("The number is negative.")
```

3. **Write a program to ask for user age and determine if they are an adult or not.**

```
age = int(input("How old are you? "))
if age >= 18:
    print("You are an adult!")
else:
    print("You are a teenager or a kid.")
```

4. **Write a Python program to create a new file and write a message into it.**

```
file = open("message.txt", "w")
file.write("Hello, welcome to file handling in Python!")
file.close()
print("File created and message written successfully.")
```

5. **Write a function to calculate the area of a rectangle.**

```
def area(length, width):
    return length * width
length = int(input("Enter length: "))
width = int(input("Enter width: "))
print("Area of the rectangle:", area(length, width))
```

6. **Write a function to check if a number is even or odd.**

```
def check_even_odd(num):
    if num % 2 == 0:
        return "Even"
    else:
        return "Odd"
num = int(input("Enter a number: "))
print("The number is:", check_even_odd(num))
```

7. **Write a function to return the square and cube of a number.**

```
def square_and_cube(n):
    return n ** 2, n ** 3
num = int(input("Enter a number: "))
sq, cb = square_and_cube(num)
print("Square:", sq, "Cube:", cb)
```

8. **Write a program using a for loop to print numbers from 1 to 10.**

```
for i in range(1, 11):
    print(i)
```

9. Write a program using a while loop to find the sum of first 10 natural numbers.

```
sum = 0
i = 1
while i <= 10:
    sum += i
    i += 1
print("Sum of first 10 natural numbers:", sum)
```

10. Write a program to print all even numbers from 1 to 20 using a for loop.

```
for i in range(2, 21, 2):
    print(i)
```

11. Write a function to calculate simple interest.

```
def simple_interest(principal, time, rate=10):
    return (principal * time * rate) / 100

p = float(input("Enter Principal Amount: "))
t = float(input("Enter Time in Years: "))
print("Simple Interest:", simple_interest(p, t))
```

4.6 File Handling using Panda Library

To store data temporarily and permanently, we use files. A file is the collection of data stored on a disk in one unit identified by filename. Python provides several built-in functions and methods for creating, opening, reading, writing, and closing files.

Pandas makes data importing and data analyzing much easier. Pandas builds on packages like NumPy and matplotlib to give us a single and convenient place for data analysis and visualization work.

4.6.1 Concept of File Handling in Python

File handling in Python involves interacting with files on your computer to read data from them or write data to them. The pandas library in Python is a powerful tool for data manipulation and analysis. When it comes to file handling, pandas make it incredibly easy to read from and write to various file formats such as CSV, Excel, JSON (Javascript Object Notation), and more.

Benefits of file handling in Python

i. **Versatility:** File handling in Python allows us to perform a wide range of operations, such as creating, reading, writing, appending, renaming and deleting files.

ii. **Flexibility:** File handling in Python is highly flexible, as it allows us to work with different file types (e.g. text files, binary files, CSV files, etc.) and to perform different operations on files (e.g. read, write, append, etc.).

iii. **User friendly**: Python provides a user-friendly interface for file handling, making it easy to create, read and manipulate files.

iv. **Cross-platform**: Python file-handling functions work across different platforms (e.g. Windows, Mac, Linux), allowing for seamless integration and compatibility.

Difficulties of file handling in Python

i. **Error-prone:** File handling operations in Python can be prone to errors, especially if the code is not carefully written or if there are issues with the file system (e.g. file permissions, file locks, etc.).

ii. **Security risks:** File handling in Python can also pose security risks, especially if the program accepts user input that can be used to access or modify sensitive files on the system.

iii. **Complexity:** File handling in Python can be complex, especially when working with more advanced file formats or operations. Careful attention must be paid to the code to ensure that files are handled properly and securely.

iv. **Performance:** File handling operations in Python can be slower than other programming languages, especially when dealing with large files or performing complex operations.

4.6.2 Concept of Mode of File Handling (Read, Write, and append a File)

Opening a file in Python

To perform any file operation, the first step is to open the file. Python's built-in **open()** function is used to open files in various modes, such as reading, writing, and appending.

Syntax : with open('Filename', 'mode') as file:

There are six different access modes that you can use when working with files:

Mode	Description	Behavior
r	Read-only mode	This mode allows you to read from the file, but you can't make any changes to it. It raises an error if the file is not existing on your system.
w	Write-only mode	This mode allows you to write to the file, but you can't read from it. It creates the file if it does not exist on your system. It will overwrite if the file is present in your system.
a	Append-only mode	This mode allows you to write to the file, but you can't read from this mode. It creates the file if it does not exist on your system. It will add new data if the file already has data.

Reading a file

Reading a file can be achieved by **file.read()** which reads the entire content of the file. We can close the file using **file.close()** which closes the file after reading it, which is necessary to free up system resources.

Example:

```
file = open("Test.txt", "r")
content = file.read()
print(content)
file.close()
```

Output:

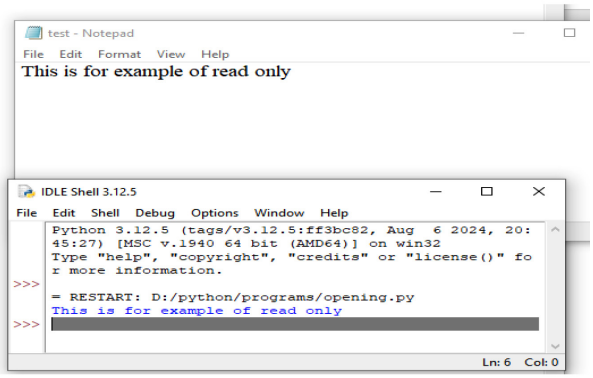


Figure 4.39

Read a file in Python

To read a file in Python, the file must first be opened in the read mode.

a) Python offers several methods of reading.

i. If you want a read-only first line only, use **readline (),** and if you want to read all the lines in the form of lines, use **readlines()**

ii. If you want to read a few characters of the file, use **read (size)**

iii. **read ()** used to read all the content at a time.

Example:

```
#read a file
nl = open('nl.txt','r') # it will open the file in read mode
nl.readline() # it will read the first line of the file
nl.read(10) # it will read the first 10 charcter of the file
nl.read() # it will return the whole text
```

Create a new file in Python

To create a new file, we have to open a file using one of the two parameters:

x: It will create a new empty text file if there does not exist any file with the same name; otherwise, it will throw an error

w: It will create a file, whether any file exists with the same or not, i.e., it will not return an error if the same file name exists.

Example:

#create a file

nl = open ('nl.txt', 'x') # It will create a new empty file but will throw an error if the same file name exists

nl = open ('nl.txt', 'w') # It will create a file but if the same file name exist it will overwrite

Writing to a file

Writing to a file is done using **file.write()** which writes the specified string to the file. If the file exists, its content is erased and new content will be added. If it doesn't exist, a new file is created.

Example:

Writing to a file in Write Mode (w)

```
file = open("Test.txt", "w")
file.write("Hello, World!")
file.close()
```

output:

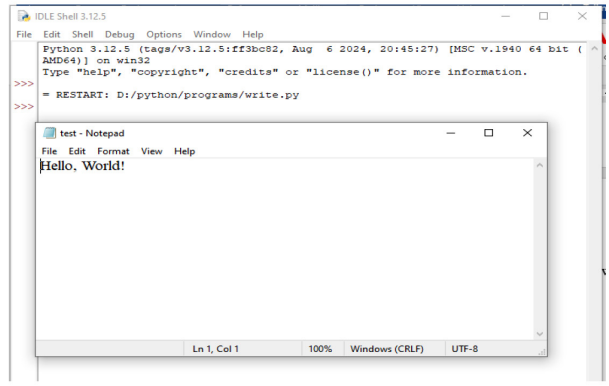


Figure 4.40

Writing to a file in Append Mode (a)

Python offers two methods to write in an existing file:

(a) write () method

It will add a single line at a time

Example:

```
with open('file.txt','w') as nl:
    nl.write('This is the first line\n')
    nl.write('This is the second line\n')
    nl.write('this is the third line')
```

(b) writelines () method

It allows to insert multiple string at a time

Example:

```
nl = open('file.txt', 'w+')
#It allows to write all the line in one go
nl.writelines(['This is the first line\n', 'This is the second line\n', 'This is the third line\n'])
nl.close()
```

Example:

```
# Python code to illustrate append() mode
file = open('geek.txt', 'a')
file.write("Python programming is fun")
file.close()
```

Output:

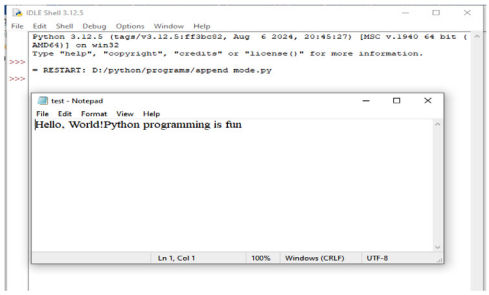


Figure 4.41

Closing a file in Python

Once you are done with all the operations, you need to close the file correctly.

- The close () commands will terminate all the resources and will free the system
- It is a good practice to close the file.

Example:

```
#close a file
nl = open('nl.txt', 'r')
print(nl.read())
nl.close()
```

4.6.3 Read and write CSV file using standard library (eg. Panda)

CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases. The structure of a CSV file is given away by its name. CSV files are normally created by programs that handle large amounts of data. Normally, CSV files use a comma to separate each specific data value. Structure is as shown below:

column 1 name, column 2 name, column 3 name

first row data 1, first row data 2, first row data 3

second row data 1, second row data 2, second row data 3

CSV files are favoured for a few key reasons:

a) **Portability**: CSV files are plain text, making them easy to open and edit in virtually any text editor or spreadsheet program (e.g., Excel, Google Sheets).

b) **Simplicity**: The structure of a CSV file is straightforward, so it's easy to generate and parse them programmatically.

c) **Wide support**: Almost every programming language, including Python, has built-in support for reading and writing CSV files.

Benefit of the csv Module

a) **Built-in and easy to use**: No additional installation is needed, and it is straightforward to implement.

b) **Flexible and adaptable**: Handles different types of delimiters, quoting, and escaping with ease, making it suitable for working with CSV files from various sources.

c) **Memory efficient**: Python's csv module reads and writes data row by row, which means it can handle large files without consuming too much memory.

Reading CSV

Reading from a CSV file is done using the reader object. The CSV file is opened as a text file with Python's built-in open() function, which returns a file object. This is then passed to the reader, which does the heavy lifting. We use following to read CVS file

import csv

There are two main ways to read CSV files in Python:

a) **Using the csv module:** This is the built-in module for working with CSV files in Python. It provides basic functionality for reading and writing CSV data.

Example using csv.reader:

```
import csv
# Open the CSV file in read mode
  with open('data.csv', 'r') as csvfile:
# Create a reader object
  csv_reader = csv.reader(csvfile)
# Iterate through the rows in the CSV file
  for row in csv_reader:
 # Access each element in the row
   print(row)
```

Output

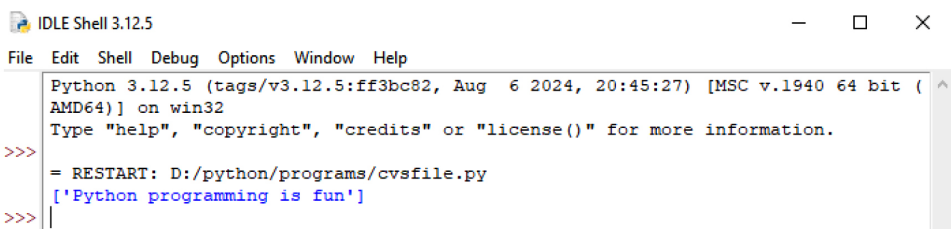


Figure 4.42

a) Using the Pandas library: Pandas is a powerful library for data analysis in Python. It offers a more convenient way to read and manipulate CSV data. We should use following syntax to read file using panda library

import pandas as pd

Let's look at how pandas are used to read data in a CSV file.

Step1: Import pandas library

import pandas as pd

Step2: Load CSV files to pandas using read_csv()

Basic Syntax: pandas.read_csv(filename, delimiter=',')

data= pd.read_csv("Salary_Data.csv")

data

	Years	Experience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0

Step3: Extract the field names.

.columns *is used to obtain the header/field names.*

data.columns

Step4: Extract the rows.

All the data of a data frame can be accessed using the field names.

data.Salary

Example

```
import pandas as pd
# Reading the CSV file
df = pd.read_csv(Data.csv')
# Displaying the content
print(df)
```

Output:

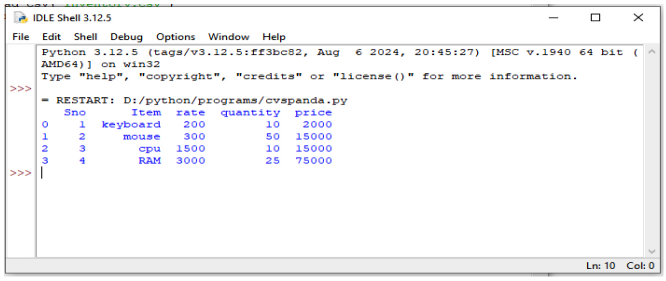


Figure 4.43

Write to a CSV file

The DataFrame supports the to_csv() method used to write a CSV file. In the pandas library, we can write a file by creating a DataFrame using the pd.DataFrame() method. This will call the to_csv() function to write people_records.csv file.

We need to follow these steps to write to a CSV file using pandas:

Step1: Import pandas library

import pandas as pd

Step2: Create a pandas dataframe using pd.DataFrame

Syntax: pd.DataFrame(data, columns)

The data parameter takes the records/observations, and the columns parameter takes the columns/field names.

header = [‘Name’, ‘M1 Score’, ‘M2 Score’]

data = [[‘Sanskar’, 62, 80], [‘Sambriddhi’, 45, 56], [‘Aurora’, 85, 98]]

data = pd.DataFrame(data, columns=header)

Step3: Write to a CSV file using to_csv()

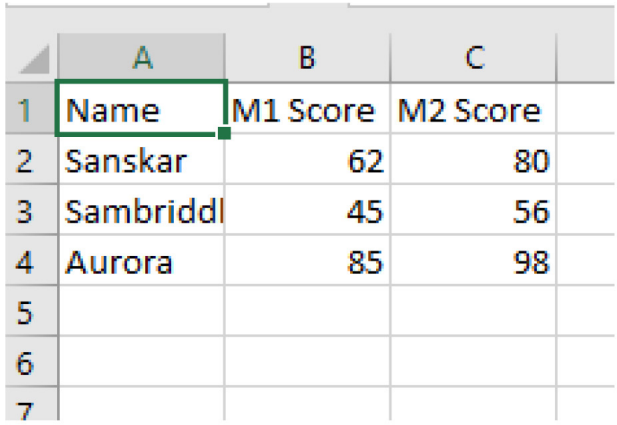
Syntax: DataFrame.to_csv(filename, sep=’,’, index=False)

**separator is ‘,’ by default.

index=False to remove the index numbers.

data.to_csv(‘Stu_data.csv’, index=False)

Output



	A	B	C
1	Name	M1 Score	M2 Score
2	Sanskar	62	80
3	Sambriddl	45	56
4	Aurora	85	98
5			
6			
7			

Figure 4.44

Example

import pandas as pd

Data to be written

data = {

```
    'Product': ['Laptop', 'Smartphone', 'Tablet'],
    'Price': [75000, 15000, 20000],
    'Quantity': [3, 10, 5]
}
# Creating a DataFrame
df = pd.DataFrame(data)
# Writing the DataFrame to a CSV file
df.to_csv('products.csv', index=False)
print("Data written to products.csv successfully.")
```

Output

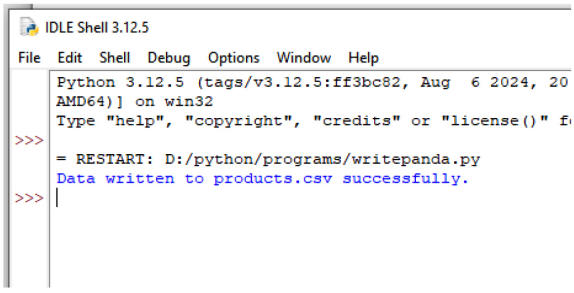


Figure 4.45

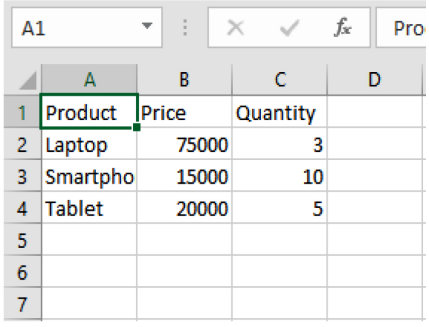


Figure 4.46

Exercise

1. Choose the correct answer.

i. Which Python library is particularly useful for simplifying file handling of structured data like CSV files?

a) math
b) random
c) pandas
d) turtle

ii. What is the primary function used in Pandas to read data from a CSV file?

a) open()
b) read_file()
c) pd.read_csv()
d) df.read_csv()

iii. When reading a CSV file with Pandas, which parameter in read_csv() is used to specify the separator between values?

a) separator
b) sep
c) delimiter
d) value_sep

iv. What attribute of a Pandas DataFrame can be used to obtain the header or field names after reading a CSV file?

a) headers
b) columns
c) fields
d) names

v. Which Pandas method is used to write a DataFrame to a CSV file?

a) write_csv()
b) to_file()
c) df.to_csv()
d) pd.write_csv()

vi. When writing a DataFrame to a CSV file using to_csv(), what does the parameter index=False do?

a) It includes the index as a column in the CSV.
b) It removes the header row from the CSV.
c) It excludes the index column from the CSV.
d) It sorts the data based on the index.

vii. **What is the default separator used by Pandas when reading or writing CSV files?**

a) semicolon (;) b) tab (\t)

c) comma (,) d) space ( )

viii. **Which mode should be used with the built-in open() function if you want to read a file?**

a) "w" b) "a" c) "r" d) "x"

ix. **Which mode, when used with the built-in open() function, will overwrite the file if it exists or create a new file if it doesn't?**

a) "r" b) "a"

c) "w" d) "x"

x. **Which Pandas function is used to create a DataFrame from a dictionary or a list of lists?**

a) read_csv() b) to_csv()

c) pd.DataFrame() d) create_df()

2. Write short answers to these questions.

i. What are the primary advantages of using the Pandas library for file handling in Python?

ii. Explain the concept of a CSV file and why it is a common format for data exchange.

iii. What is the first step you need to take to use the Pandas library in your Python script for file handling?

iv. Describe the basic syntax for reading a CSV file into a Pandas DataFrame.

v. How can you access a specific column of data after reading a CSV file into a Pandas DataFrame? Provide an example.

vi. Explain the basic syntax for writing a Pandas DataFrame to a CSV file.

vii. What happens if you try to open a non-existent file in read ("r") mode using the built-in open() function?

viii. Explain the difference between the write ("w") mode and the append ("a") mode when opening a file.

ix. Why is it important to close a file after you are done with read or write operations (even when using Pandas)?

3. Write long answers to these questions.

i. Write the steps to read data from a CSV file using Pandas in Python. Also, give a simple code to:

- Import the Pandas library
- Read the CSV file

ii. Imagine you have a CSV file named "student_data.csv" with columns "Name", "Age", and "Grade". Write a Python program using the Pandas library to:

a. Read the data from the "student_data.csv" file into a DataFrame.

b. Print the first 5 rows of the DataFrame.

c. Calculate the average age of the students.

d. Create a new DataFrame containing only the students with a "Grade" of "A".

4.7 Introduction to Data Visualization

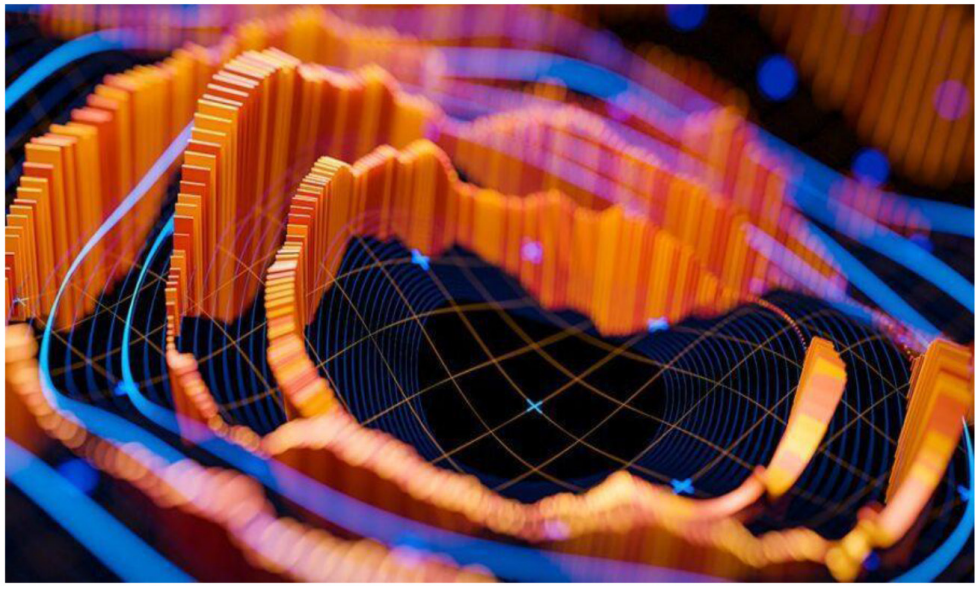


Figure 4.47

Data visualization means showing data using charts or graphs to understand it better. It helps us see patterns, trends, and connections in a simple way.

Python has many useful libraries to make these charts. You can create simple or advanced graphs easily using tools like **Matplotlib** or **Plotly**

Importance of Data Visualization

Data visualization helps people understand large amounts of data easily by turning it into charts and graphs. This makes it easier to see trends, patterns, and new ideas.

Today, businesses and professionals use data to make better decisions. Since a huge amount of data is created every day, visualization helps us make sense of it and share our ideas clearly.

Python has different libraries for data visualization. Each library has its own features and can create different types of graphs.

Some popular Python libraries are:

i. **Matplotlib** – Good for simple graphs like bar charts and line graphs.
ii. **Seaborn** – Makes more beautiful and colorful charts easily.
iii. **Plotly** – Used for interactive and dynamic graphs.

Matplotlib

Matplotlib is the most popular Python plotting library. It is a low-level library with a Matlab-like interface that offers lots of freedom at the cost of having to write more code. To install Matplotlib, pip, and conda can be used, and should install using command prompt.

pip install matplotlib

or

conda install matplotlib

Matplotlib is specifically suitable for creating basic graphs like line charts, bar charts, histograms, etc. It can be imported by typing:

import matplotlib.pyplot as plt

Scatter plot

Dots are used to represent relationships between variables in scatter plots to observe relationships between variables. To draw a scatter plot, the matplotlib library provides the scatter() method.

Example:

```
#Importing the libraries
import pandas as pda
import matplotlib.pyplot as plt
# Reading the dataset
dataset = pda.read_csv("Stu_data.csv")
plt.scatter(dataset['Name'],
dataset['Marks'])
plt.title("Scatter Plot")
plt.xlabel('Name')
plt.ylabel('Marks')
plt.show()
```

Output:

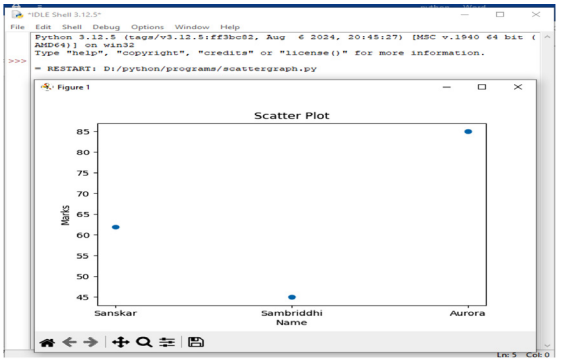


Figure 4.48

Bar chart

To represent a data category, a bar plot or bar chart uses rectangular bars whose lengths and heights correspond to the data values they represent.

Example

```
# Importing the libraries
import pandas as pda
import matplotlib.pyplot as plt
# Reading the database
data = pda.read_csv("tips.csv")
plt.bar(data['total_bill'],
data['day'])
plt.title("Bar Chart")
plt.xlabel('Day')
plt.ylabel('Tip')
plt.show()
```

Output:

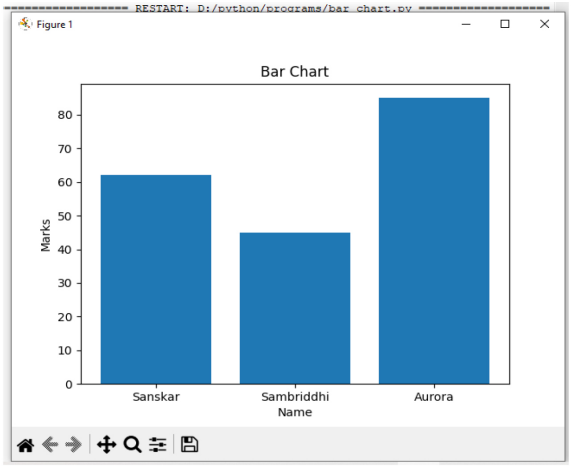


Figure 4.49

Seaborn

Seaborn is a Python library used to make beautiful and informative charts. It is built on top of **Matplotlib** and works well with **Pandas** data.

Seaborn is great for creating charts that show patterns and relationships in data, such as:

i. Line plots

ii. Bar plots

iii. Heatmaps

It also helps make complex charts using fewer lines of code.

To install Seaborn, use this command in the command prompt:

pip install Seaborn

It works best in **Jupyter Notebook** or **IPython**, which show the charts more clearly.

Line Plot

To plot a line plot in Seaborn, use the lineplot() method. It is also possible to pass just the data argument in this case.

Example:

```
# Importing the libraries
import pandas as pda
import seaborn as sn
import matplotlib.pyplot as plt
# Reading the database
dataset = pda.read_csv("Stu_data.csv")
sn.lineplot(x='Name', y='Marks', data=dataset)
plt.show()
```

Output:

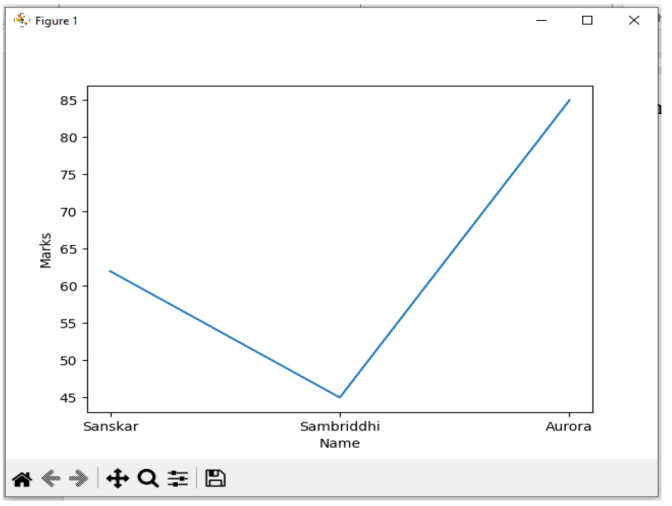


Figure 4.50

Plotly

Plotly is an open-source Python library used to make interactive charts. It can create different types of graphs like:

i. 3D charts

ii. Scientific and statistical charts

iii. Financial charts

You can view Plotly graphs in Jupyter notebooks, web browsers, or save them as HTML files. It also lets you click, zoom, and edit charts easily. To use Plotly we should install it by using following command

pip install plotly

Scatter Plot Plotly's scatter() method can be used to create scatter plots. It is also necessary to include an additional data argument, like Seaborn. T

```
import pandas as pda

import plotly.express

# we are reading the csv dataset through pandas

dataset = pda.read_csv("tips.csv")

graph = plotly.express.scatter(dataset, x="total_bill", y="size", color='smoker')

graph.show()
```

Output:

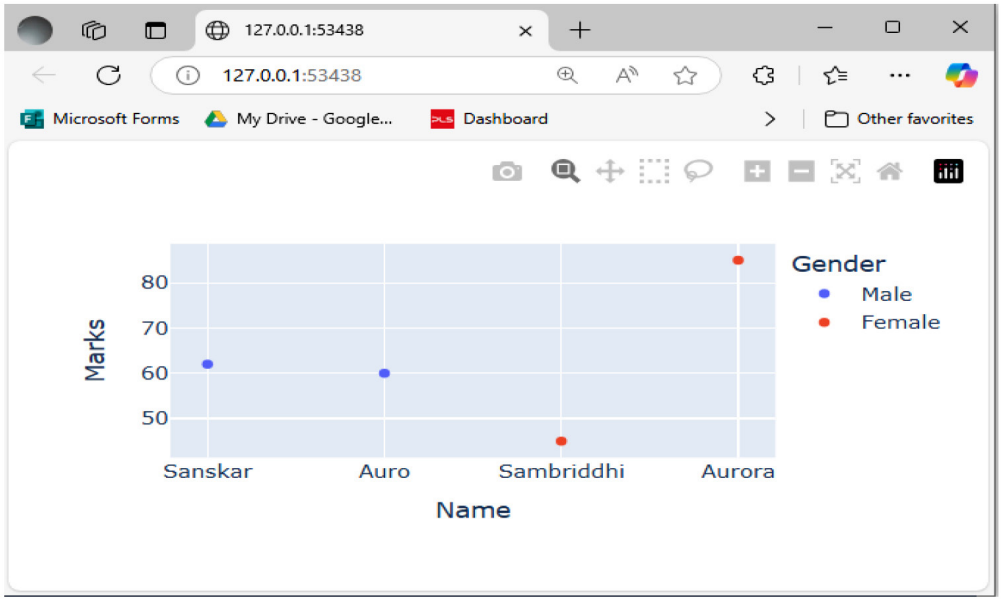


Figure 4.51

Line chart

In Plotly, line plots are much more accessible and illustrious additions that assemble easy-to-style statistics from various data types**.** Each position of data is represented as a vertex with px. Line.

Example:

```
# importing the library
import plotly.express as px
import pandas as pda
# we are reading the database
data = pda.read_csv("stu_data.
csv")
# we are plotting the scatter
chart
fig = px.line(data, y='Name',
color='Gender')
fig.show()
```

Output:

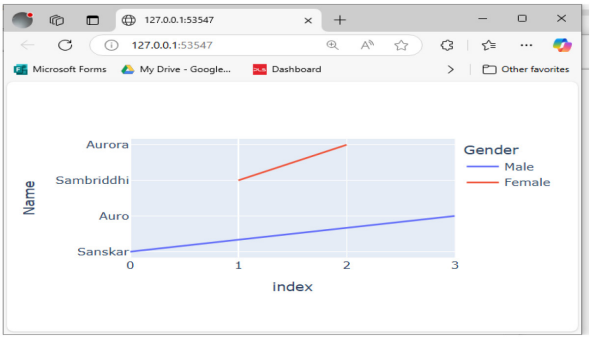


Figure 4.52

Bar chart

With plotly.express, you can create bar charts using the bar() method.

```
# importing the libraries

import plotly.express as px

import pandas as pd

# reading the database

data = pd.read_csv(“Stu_data.csv”)

fig = px.bar(data, x=’Name’, y=’Marks’, color=’Gender’)

# showing the plot

fig.show()
```

Output:

Figure 4.53

A. Solved examples:

1. Create a new text file and write content

```
file = open("example.txt", "w")
file.write("Hello, this is a new file created using Python file handling.")
file.close()
print("File created and content written successfully.")
```

2. Append new content to an existing file

```
file = open("example.txt", "a")
file.write("\nThis is an appended line.")
file.close()
print("New content appended successfully.")
```

3. Read content from a file

```
file = open("example.txt", "r")
content = file.read()
print("File content:\n", content)
file.close()
```

4. Read the first 10 characters of a file

```
file = open("example.txt", "r")
print("First 10 characters:", file.read(10))
file.close()
```

5. Read a CSV file using Pandas

```
import pandas as pd
df = pd.read_csv("sample_data.csv")
print(df)
```

6. Count the number of rows in a CSV file

```
import pandas as pd
df = pd.read_csv("sample_data.csv")
print("Number of rows:", len(df))
```

7. Pie chart using Matplotlib

```
import matplotlib.pyplot as plt
labels = ['A', 'B', 'C', 'D']
sizes = [20, 30, 25, 25]
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.title("Pie Chart Example")
plt.show()
```

9. Bar chart using Plotly

```
import plotly.express as px
import pandas as pd
data = pd.DataFrame({'Category': ['A', 'B', 'C', 'D'], 'Value': [10, 20, 30, 40]})
fig = px.bar(data, x='Category', y='Value', title="Bar Chart Example")
fig.show()
```

10. Save a DataFrame as CSV using Pandas

```
import pandas as pd
data = {
    'Product': ['Laptop', 'Phone', 'Tablet'],
    'Price': [700, 300, 200],
    'Stock': [50, 100, 80]
}
df = pd.DataFrame(data)
df.to_csv('products.csv', index=False)
print("Data saved to 'products.csv' successfully.")
```

B. Code Practice:

1. Write a Python program that asks any two numbers and displays the greater one.

```
# Ask for two numbers from the user
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
# Compare and display the greater number
if num1 > num2:
print("The greater number is:", num1)
elif num2 > num1:
```

```
print(“The greater number is:”, num2)
else:
print(“Both numbers are equal.”)
```

2. Write a Python program that asks the length, breadth, and height of a room. Create a user-defined function to calculate the area of the floor. Use another function to calculate the volume of the room. Hint: Area = Length × Breadth, Volume = Length × Breadth × Height

```
# Function to calculate area of the floor
def calculate_area(length, breadth):
    return length * breadth
# Function to calculate volume of the room
def calculate_volume(length, breadth, height):
    return length * breadth * height
# Taking input from the user
length = float(input("Enter the length of the room (in meters): "))
breadth = float(input("Enter the breadth of the room (in meters): "))
height = float(input("Enter the height of the room (in meters): "))
# Calculating area and volume
area = calculate_area(length, breadth)
volume = calculate_volume(length, breadth, height)
# Displaying results
print("Area of the floor:", area, "square meters")
print("Volume of the room:", volume, "cubic meters")
```

3. Write a Python program that asks for the length in feet and converts it to inches. Hint: 1 foot = 12 inches

```
# Function to convert feet to inches
def feet_to_inches(feet):
    return feet * 12
```

```
# Ask the user for input
feet = float(input("Enter the length in feet: "))
# Convert and display the result
inches = feet_to_inches(feet)
print("Length in inches:", inches)
```

4. Write a Python program that asks three different numbers and calculates the product and average. Use a function for product and a subprogram for average.

```
# Function to calculate product
def find_product(a, b, c):
    return a * b * c

# Subprogram to calculate average
def show_average(x, y, z):
    average = (x + y + z) / 3
    print("The average is", average)

# Ask user for three numbers
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

# Call function and subprogram
product = find_product(num1, num2, num3)
print("The product is", product)
show_average(num1, num2, num3)
```

5. Write a program to calculate the area of a square using a function. The user should input the side length.

```
# Function to calculate area of a square
def find_area(side):
    area = side * side
    return area
```

```
# Ask user to enter the side of the square
side_length = float(input("Enter the side length of the square: "))

# Call the function and display the result
area = find_area(side_length)
print("The area of the square is", area)
```

6. Write a program that calculates the Total Surface Area of a cuboid using a function. ***Hint: TSA = 2(lb + bh + hl)***

```
    # Function to calculate Total Surface Area of a cuboid
def total_surface_area(length, breadth, height):
    tsa = 2 * (length * breadth + breadth * height + height * length)
    return tsa

# Ask user to enter length, breadth, and height
l = float(input("Enter the length of the cuboid: "))
b = float(input("Enter the breadth of the cuboid: "))
h = float(input("Enter the height of the cuboid: "))

# Call the function and display the result
tsa = total_surface_area(l, b, h)
print("The Total Surface Area of the cuboid is", tsa)
```

7. Write a Python program to calculate the perimeter of a square using a subprogram. ***Hint: Perimeter = 4 × L***

```
# Subprogram to calculate and display perimeter of a square
def show_perimeter(side):
    perimeter = 4 * side
    print("The perimeter of the square is", perimeter)

# Ask user to enter the side length
side_length = float(input("Enter the side length of the square: "))

# Call the subprogram
show_perimeter(side_length)
```

8. **Write a Python program to calculate the circumference of a circle using a subprogram. *Hint: Circumference = 2πr***

```
import math   # To use the value of pi (π)

# Subprogram to calculate and display circumference

def show_circumference(radius):

circumference = 2 * math.pi * radius
print("The circumference of the circle is", circumference)

# Ask user to enter the radius
r = float(input("Enter the radius of the circle: "))

# Call the subprogram
show_circumference(r)
```

9. **Write a program to calculate the total surface area of a sphere using a subprogram. Hint: TSA = $4\pi r^2$**

```
import math  # To use the value of pi (π)


# Subprogram to calculate and display TSA of a sphere

def show_surface_area(radius):

    tsa = 4 * math.pi * radius * radius

    print("The Total Surface Area of the sphere is", tsa)


# Ask user to enter the radius

r = float(input("Enter the radius of the sphere: "))


# Call the subprogram
show_surface_area(r)
```

10. Write a program to calculate the curved surface area (CSA) of a cuboid.
Hint: CSA = 2h(l + b)

```
# Subprogram to calculate and display CSA of a cuboid
def show_csa(length, breadth, height):
    csa = 2 * height * (length + breadth)
    print("The Curved Surface Area of the cuboid is", csa)

# Ask user to enter length, breadth, and height
l = float(input("Enter the length of the cuboid: "))
b = float(input("Enter the breadth of the cuboid: "))
h = float(input("Enter the height of the cuboid: "))

# Call the subprogram
show_csa(l, b, h)
```

11. Write a program using a subprogram to generate the series: 2, 3, 5, 8, 13, 21, 34 (up to 10 terms), and calculate the sum of the series.

```
# Program to generate the series and calculate its sum

# Initialize first two terms
a = 2
b = 3
total = a + b

print("Series:")
print(a, b, end=" ")

# Generate the remaining 8 terms
for _ in range(8):
    next_term = a + b
    print(next_term, end=" ")
    total += next_term
    a = b
    b = next_term

# Display the sum
print("\nSum of the series is:", total)
```

12. Write a Python program to find the sum of digits of a number using a user-defined function. For example, if the user enters 123, the result should be 6.

```
# Function to find sum of digits
def sum_of_digits(num):
    total = 0
    while num > 0:
        digit = num % 10
        total = total + digit
        num = num // 10
    return total
# Ask user for input
number = int(input("Enter a number: "))
# Call the function and display result
result = sum_of_digits(number)
        print("Sum of digits is:", result)
```

13. Write a Python program to display the multiplication table of a number up to 10 using a user-defined function.

```
# Function to display multiplication table
def show_table(num):
    print("Multiplication Table of", num)
    for i in range(1, 11):
        print(num, "x", i, "=", num * i)
# Ask user for input
number = int(input("Enter a number: "))
# Call the function
show_table(number)
```

14. Write a Python program to find the factorial of a number using a user-defined function. Example: 5! = 5 × 4 × 3 × 2 × 1 = 120

```
# Function to calculate factorial

def find_factorial(n):
    fact = 1
    for i in range(1, n + 1):
        fact = fact * i
    return fact
# Ask user for input
number = int(input("Enter a number: "))
# Call the function and show result
result = find_factorial(number)
print("Factorial of", number, "is", result)
```

15. Write a Python program to create a file named student.txt and store the name and class of 3 students.

```
file = open("student.txt", "w")
for i in range(3):
    name = input("Enter name of student: ")
    clas = input("Enter class of student: ")
    file.write(name + "," + clas + "\n")

file.close()
print("Data written to student.txt")
```

16. **Write a Python program to read and display the content of student.txt.**

```
file = open("student.txt", "r")
print("Contents of student.txt:")
for line in file:
    print(line.strip())
file.close()
```

17. **Write a Python program to add (append) one more student's name and class to the existing file student.txt.**

```
file = open("student.txt", "a")
name = input("Enter name of new student: ")
clas = input("Enter class of new student: ")
file.write(name + "," + clas + "\n")
file.close()
print("Data added to student.txt")
```

18. **Write a Python program to generate this series using a function: 2 3 5 8 13 21 34 (up to 10 terms) and also, calculate and display the sum of the series.**

```
# Function to generate the series
def generate_series(n):
series = [2, 3]  # Starting values
for i in range(2, n):
next_number = series[i-1] + series[i-2]
series.append(next_number)
return series
```

```
# Generate and display the series
terms = 10
series = generate_series(terms)
series_sum = sum(series)
print("Generated Series:")
print(series)
print("Sum of the series:", series_sum)
```

19. Write a Python program to input two numbers and calculate: Sum, Product, Difference. Use three separate functions for each calculation.

```
# Function to calculate sum
def find_sum(a, b):
    return a + b
# Function to calculate product
def find_product(a, b):
    return a * b
# Function to calculate difference
def find_difference(a, b):
    return a - b
# Input from user
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
# Perform calculations
sum_result = find_sum(num1, num2)
product_result = find_product(num1, num2)
difference_result = find_difference(num1, num2)
# Display the results
```

```
print("Sum:", sum_result)
print("Product:", product_result)
print("Difference:", difference_result)
```

20. A file named "employee.csv" stores Name, Address, Gender, and Salary. Write a Python program to show the records of employees who earn more than 75,000.

```
import pandas as pd
# Read the CSV file
df = pd.read_csv("employee.csv")
# Filter employees with salary > 75000
high_salary_df = df[df['Salary'] > 75000]
# Display the filtered records
print("Employees with salary more than 75,000:")
print(high_salary_df)
```

21. A file named "result.csv" stores data with columns: Symbol No, Name, Gender, English, Nepali, Maths, Computer. Write a Python program to display the records of students: Whose gender is 'M' and who scored more than 85 in Computer

```
import pandas as pd
# Read the CSV file
df = pd.read_csv("result.csv")
# Filter the records where Gender is 'M' and Computer marks > 85
filtered_df = df[(df['Gender'] == 'M') & (df['Computer'] > 85)]
# Display the filtered records
print("Students with Gender 'M' and Computer marks > 85:")
print(filtered_df)
```

22. A file "employee.csv" contains data with fields: Name, Address, Gender, and Salary. Write a Python program to display the data of employees with salary more than 50,000.

```
import pandas as pd
# Read the CSV file
df = pd.read_csv("employee.csv")
# Filter employees with salary greater than 50000
high_salary_df = df[df['Salary'] > 50000]
# Display the filtered data
print("Employees with salary more than 50,000:")
print(high_salary_df)
```

23. A file named "record.csv" contains columns: Roll No, Name, Gender, English, Nepali, Maths, Computer. Write a Python program to display the records of students whose gender is 'F' and who scored more than 90 in Computer

```
import pandas as pd
# Read the CSV file
df = pd.read_csv("record.csv")
# Filter: Gender is 'F' and Computer marks > 90
filtered_df = df[(df['Gender'] == 'F') & (df['Computer'] > 90)]
# Display the filtered records
print("Female students who scored more than 90 in Computer:")
print(filtered_df)
```

Exercise

1. Choose the correct answer.

i. What is the primary goal of data visualization?

a) To store data efficiently.
b) To perform complex calculations on data.
c) To understand data through visual context.
d) To secure data from unauthorized access.

ii. Which of the following is a popular Python library for creating basic graphs like line charts and bar charts?

a) Seaborn
b) Plotly
c) Matplotlib
d) Pandas

iii. Which type of plot uses dots to represent relationships between variables?

a) Bar chart
b) Line chart
c) Scatter plot
d) Pie plot

iv. What type of chart uses rectangular bars to represent data categories?

a) Scatter plot
b) Line chart
c) Bar chart
d) Pie plot

v. Which data visualization library in Python is built on Matplotlib and offers more advanced statistical visualizations?

a) Plotly
b) Pandas
c) Seaborn
d) GGPlot

vi. Which Plotly method is used to create a scatter plot?

a) scatter()
b) line()
c) bar()
d) pie()

vii. Which Plotly Express function is used to create a line chart?

a) px.scatter()
b) px.line()
c) px.bar()
d) px.pie()

viii. **Which Matplotlib function is commonly used to create a pie chart?**

a) plt.scatter() b) plt.plot()

c) plt.bar() d) plt.pie()

ix. **Which Plotly Express function is used to create a bar chart?**

a) px.scatter() b) px.line()

c) px.bar() d) px.histogram()

2. Write short answers to these questions.

a) Define data visualization in your own words.

b) Why is data visualization considered important for businesses and analysts?

c) Name three popular Python libraries for data visualization.

d) What is the key characteristic of Matplotlib that offers both freedom and the need for more code?

e) What type of data is typically represented using a bar chart?

f) What is Seaborn built upon, and what type of visualizations does it primarily focus on?

g) What is a key feature of Plotly that distinguishes it from Matplotlib?

h) In Matplotlib, what is the role of plt.xlabel() and plt.ylabel()?

i) What type of data is best represented using a pie plot?

j) What is the purpose of the color argument in Plotly Express plotting functions?

3. Write long answers to these questions.

i. Explain the importance of data visualization in the process of data analysis.

ii. Compare and contrast the Matplotlib and Plotly libraries for data visualization in Python. Discuss their key features, strengths, and

weaknesses, particularly in terms of ease of use, interactivity, and the level of customization they offer for creating line charts, pie plots, and bar graphs.

iii. Explain how to create a basic line chart**,** bar graph**,** and pie chart using the Matplotlib library in Python. In your answer, include the import statement, example data, key functions used to create and display each chart.

iv. Imagine you have a Pandas DataFrame containing sales data for different product categories over the past year. The DataFrame has columns 'Month', 'Category', and 'Sales'. Outline the steps you would take and the Python code (using either Matplotlib or Plotly) to create the following visualizations to analyze this data:

a) A line chart showing the total sales trend over the months for all categories.

b) A bar graph comparing the total sales for each product category for the entire year.

c) A pie plot showing the percentage contribution of each product category to the total sales for the entire year.

4. Practical Questions:

i. Write a Python program to take two numbers as input and print their sum, difference, product, and quotient.

ii. Write a program that takes a user's name as input and prints a greeting message using the print() function.

iii. Create a Python program that takes an integer input from the user and checks whether it is positive, negative, or zero using an if-elif-else statement.

iv. Write a program that accepts two numbers from the user and swaps their values without using a third variable.

v. Create a program that takes a sentence as input and counts the number of vowels in it.

vi. Write a Python program to create a new file called **"data.txt"** and write the sentence **"Hello, this is a test file."** into it.

vii. Modify the above program to append the sentence **"This is an appended line."** to the same file.

viii. Write a program that reads a file and prints its content to the console.

ix. Write a Python program using the **csv** module to create a CSV file **"students.csv"** with columns **"Name"** and **"Marks"**, and add three student records.

x. Write a program that reads data from **"students.csv"** and displays the content.

xi. Using Pandas, read a CSV file and print the first five rows.

xii. Write a Pandas program to create a **DataFrame** from a dictionary and save it as a

xiii. Using **Pandas and Matplotlib**, read data from a CSV file and plot a pie chart for product sales.

xiv. Write a Python function that takes two numbers as input and returns their sum.

xv. Create a function that calculates the area of a circle given the radius as a parameter.

xvi. Write a function that checks if a number is prime or not.

xvii. Write a function that takes a list of numbers as input and returns the maximum number from the list.

xviii. Write a program that reads a file and handles the exception if the file does not exist.

Unit 5

AI and Contemporary Technologies

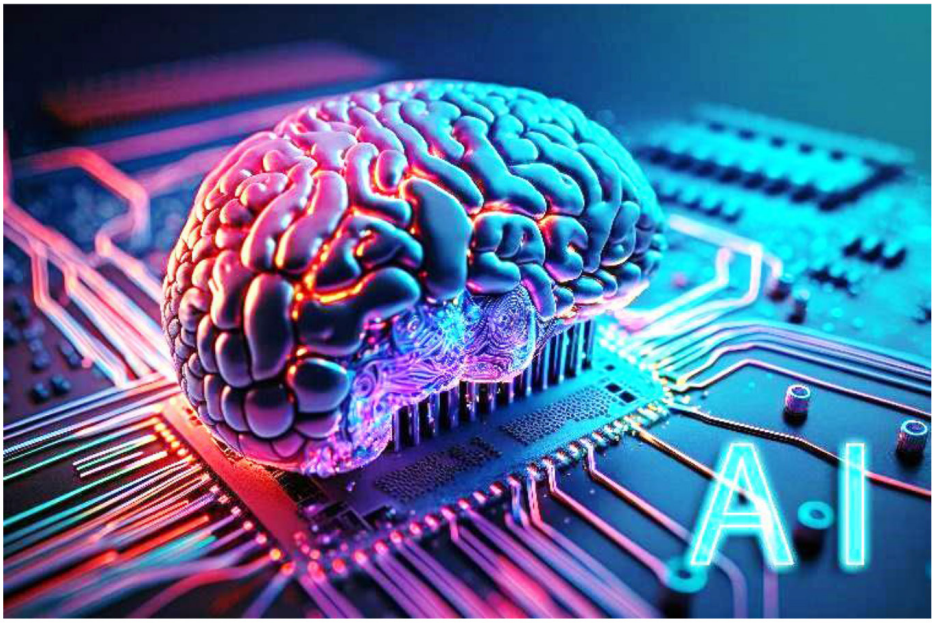


Figure 5.1

Let's think.

1. Can machines ever think like humans? "Imagine a robot playing badminton. It starts copying human player, then learns from its mistakes and learns step by step just like you!"
2. Have you ever talked with large language models (LLMs) like ChatGPT or Deepseek in your own language?
3. Where do you think your photos, videos, and documents go when you save them "in the cloud"?

The advancement of technologies has developed several machines that can learn from their past experiences and learn from their surroundings. They can respond to humans in natural languages. Robots help doctors in performing critical surgeries; virtual reality helps to experience the world that is not real.

A person can store their data and information over the internet and can access them from anywhere anytime. These technologies, like **Artificial Intelligence (AI),** robotics, virtual reality, cloud computing, and the **Internet of Things (IoT**), are changing the way we live, work, and perform daily activities.

5.1 Artificial Intelligence and Machine Learning

Artificial Intelligence (AI) is a part of computer science that focuses on creating machines that can think and act like humans. These machines are programmed to copy human intelligence, such as learning, reasoning, solving problems, planning, understanding language, and even moving objects. AI systems are designed to show at least some of these human-like abilities.

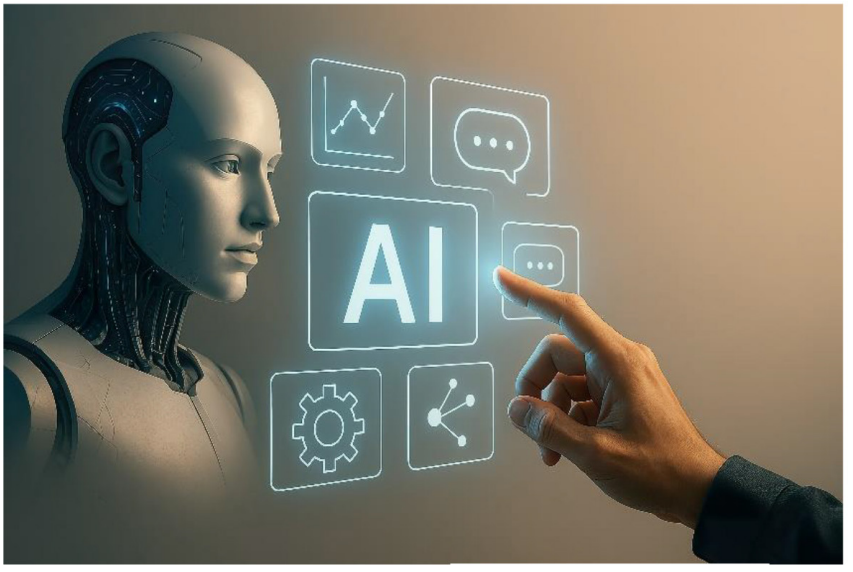


Figure 5.2

Usually, computers are used to process and store data. But with AI, computers can do much more. They can learn from past experiences, take different kinds of input, and perform tasks like humans. For example, AI is used in chess-playing computers,

self-driving cars, language translation, face detection, and answering questions. Computers with AI can analyze large amounts of data, find patterns, and use this information to complete specific tasks.

In the past, computers could only follow strict instructions given to them. But now computers can learn from past experiences; this concept is known as **machine learning**. Machine learning (ML) is a very important part of **artificial intelligence** (AI). Machine Learning helps computers learn from the data and improve their performance over time without being directly programmed. AI and ML help computers to analyze large volumes of data, find patterns, and make decisions by themselves. For example, instead of giving a computer step-by-step instructions to identify a dog or cat, we give it many examples, and it learns how to distinguish between a dog and a cat on its own. This process of learning is called machine learning, which makes AI systems more powerful and flexible.

AI and ML are closely related concepts. AI is the broad term that deals with making machines smart, and ML is the subset of AI that helps many AI systems to learn and improve. Machine learning gives AI the ability to become better with experience, just like humans learn from past experiences and practices. For example, Google maps gives you faster routes using traffic data and is an example of AI using machine learning. Together AI and ML have a greater scope and are used in various fields

Comparison between Machine Learning and Artificial Intelligence

Artificial Intelligence	Machine Learning
AI is the branch of computer science that makes computer think and behave like human.	Machine Learning is the subset of AI that focuses on making computers learn from data sets.
AI uses reasoning, problem-solving, natural language processing and more.	ML focus on data patterns and predictions based on data sets.
AI includes robots, chatbots and smart assistants.	ML includes things like email spam filters, product suggestions.
Example: A robot serving in hotel	Example: A recommendation system in YouTube.

Applications of AI:

a. Gaming

Gaming is one of the major applications of AI. AI plays an important role in modern video games by providing dynamic and engaging experience. It can play games like **chess, poker, football,** etc. with humans. It helps the computer think about many possible moves and make **smart decisions** to win the game.

Figure 5.2.1

b. Natural Language Processing (NLP)

AI helps computers understand and respond in human language. You can talk to smart assistants like **Siri or Alexa**, and they reply like a person. It also lets you control devices using your voice in your own language.

c. Automated grading

Automated grading means using AI tools to check and score students' work, such as tests, assignments, and essays. For example, tools like **Grammarly** check grammar and spelling, while tools like **Google Forms** and **Quillionz** can automatically grade quizzes and give instant feedback. These tools help teachers save time and give students quick and fair results. It also helps in improving writing and learning through suggestions and corrections.

d. Personalized learning

We know that every person learns differently. In traditional teaching, it's hard to give special attention to each student. But with AI tools like Duolingo, students can learn at their own speed and based on their interests. Instead of everyone studying the same thing in the same way, each student can have a unique learning plan.

e. Expert systems

Expert systems are smart computer programs that act like human experts in a specific field. They use stored knowledge and rules to solve problems or give advice. For example, doctors use expert systems to help find out what disease a patient might have and suggest treatments.

f. Vision systems

A computer vision system is an AI technology that can understand and analyse pictures or videos. This system uses cameras and various algorithms to **recognise, read,** and **analyse** images. For example, face lock in smartphones and facial recognition software to identify criminals.

g. Handwriting recognition

Handwriting recognition is an AI technology that can recognise and read handwriting from handwritten text from paper or a screen and convert it into digital text that can be edited on a computer.

h. Intelligent robots

Robots with AI can do tasks given by humans. They use sensors to detect things like light, heat, sound, and movement. They can learn from their mistakes and adapt to new environments. For example, robots are used in factories, hospitals, and even in space exploration.

Application of Machine Learning

a. Email spam detection

ML helps detect unwanted or dangerous emails and sends them to the spam folder instead of the inbox.

b. Speech recognition

ML allows computers to understand human speech. It changes voice into text and helps tools like Siri respond to your voice.

c. Image recognition

ML helps in face unlock on phones. It can identify people or objects in photos. It is also used in tagging photos and medical image checks.

d. Self-driving cars

These cars use ML to see roads, signs, people, and other vehicles. They make smart driving decisions like turning or stopping.

e. Chatbots and virtual assistants

Businesses use ML chatbots (like Alexa or Siri) to talk with customers. They learn from conversations to give better answers.

f. Fraud detection

Banks use ML to find fake or strange activities in accounts. It helps protect money and stop online theft.

g. Social media feeds

ML controls what you see on apps like Facebook or TikTok. It shows content based on what you like and keeps you interested.

h. Dark web monitoring

The dark web is a hidden part of the internet. ML can help find illegal activities there to reduce crime.

i. Cybersecurity

ML can protect systems by finding and stopping cyber-attacks like hacking or phishing.

Activity 1:

Create a list of recent AI tools (for mobile and desktop) and emerging technology with their key features, developers and applications. Also present it in your class.

S.N.	AI tool	Developer	Key Features	Application
1.				
2.				
3.				
4.				
5.				

Do you know?

Some AI robots can recognize emotions from your face and voice and respond with empathy just like a human! AI is used in video games to make computer players smarter and more challenging!

5.2 Learning Techniques

A modern computer can learn from its past experiences as well as the data provided to it. It can analyse the data and make some conclusions and decisions. There are two types of learning techniques: **supervised** and **unsupervised**.

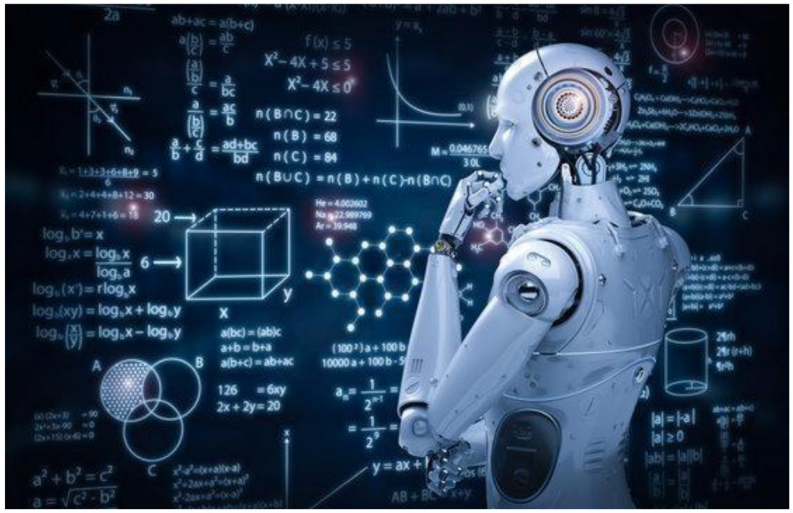


Figure 5.3

i. **Supervised learning technique:** As the name suggests, in supervised learning techniques, the machine is given some set of data with guidance. The machine is trained using labeled data. The label data means data with clear answers or tags. For example, a computer is trained with the images of a tiger and a lion to distinguish them. If new images are given to it, it will recognize tigers and lions.

ii. **Unsupervised learning technique:** The learning technique where a computer is not provided with labelled data but has to find patterns or groups by itself is

known as unsupervised learning. For example, if you give pictures of a tiger and a lion without labels, it might group them into two categories based on their appearance and characteristics.

Difference between supervised and unsupervised learning:

Supervised Learning	**Unsupervised Learning**
The computer learns from labelled data.	The computer learns from unlabelled data.
It knows the correct output while training	It tries to find patterns or groups in the data.
It is like learning with teachers who give the right answers.	It is like learning without a teacher by exploring the materials.
Example: Email is classified as spam or not spam	Example: Grouping customers based on shopping pattern

Activity 1.2:

Visit **https://teachablemachine.withgoogle.com/** website and create a machine model. Upload 5 different images of cats and 5 different images of dogs and train your model. After training the model, upload different new images of cats and dogs to check whether it is able to distinguish among cats and dogs or not. Also present your learning module in your class.

Do you know?

When you shop online, AI recommends products based on your browsing habits it's watching what you like!

Machine Learning helps self-driving cars understand traffic, recognize signs, and avoid accidents.

5.3 AI in Robotics, Simulation of Simple Robotic Tasks

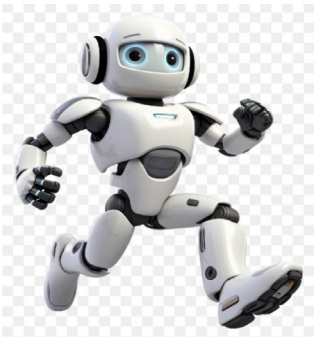


Figure 5.4

Artificial Intelligence (AI) in robotics is the study of designing, building, and operating robots. A robot consists of special sensors like cameras and microphones to take input from its surroundings to perform complex tasks automatically. Robots are especially useful for jobs that are too dangerous, difficult, or tedious for humans.

For example, they can be used for bomb disposal, exploring the ocean, fighting fires, space exploration, and working in nuclear or chemical environments.

Today, robots are used in many areas, such as manufacturing, healthcare, agriculture, and even in our homes (like robotic vacuum cleaners). They help make work faster, safer, and more efficient. As technology advances, robots are becoming smarter and more capable, changing the way we live and work.

Activity 1.3:

Simulation of simple robotic task using scratch: Develop a simple programme that makes the robot move certain steps and display different messages. (You can use various online simulators like **roboblocky.com**).

Follow the given steps to create a simple robotic task.

Step 1: Visit the website **https://scratch.mit.edu/**

Step 2: Delete the default sprite. Choose a robot sprite with an appropriate backdrop.

Step 3: Drag the following code and click on the green flag.

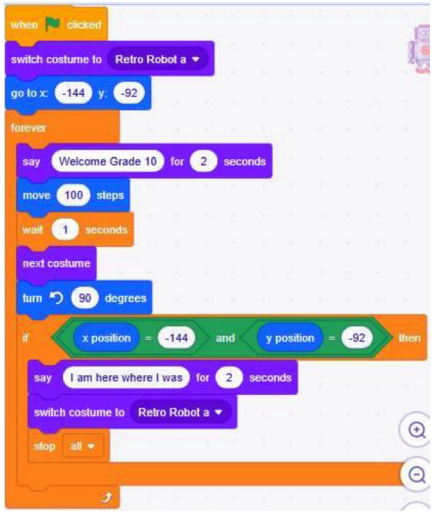


Figure 5.5 Block Code

Step 4: You will see the following result:

Figure 5.6

Activity 1.4:

Use an online robot simulator platform (like **roboblocky.com**). Create a simple robot that moves in the shape of a perfect square.

5.4 Generative AI (Gen AI)

Fig 5.7

The earlier computers could only respond with information that was already stored in them. But now, with **Generative AI**, computers can create their own content. Generative AI is the computer technology which can create new content in various forms such as **text, pictures, music or videos**. It learns from large amounts of data and then uses that learning to produce original content. Every time you ask for something, the response can be unique and different. The popular generative AI are discussed below.

5.4.1. Copilot: Copilot is the generative AI developed by Microsoft that specially help in writing codes and texts. It works inside the Microsoft apps like Word, Excel and coding tools. It makes programming easier especially for beginners.

5.4.2. ChatGPT: ChatGPT is the powerful AI **chat*bot*** developed by **Open AI**. Initially it was text based chatbot which used to generate the texts on the basis of prompts. But the advance form of ChatGPT can process **natural language, files, images** and generate response in the same way.

5.4.3. Gemini: Gemini is Google's advanced AI chatbot. Think of it as a smart assistant that helps you use Artificial Intelligence (AI) across many Google tools you might already know, like **Google Docs** and **Gmail**.

The table below compares three popular AI tools Copilot, ChatGPT, and Gemini based on their developers and the CEOs of the companies behind them.

Tool	Developed By	Company	CEO of Company
Copilot	OpenAI + GitHub	Microsoft (integrated with GitHub)	Satya Nadella
ChatGPT	OpenAI	OpenAI	Sam Altman
Gemini	Google Deep-Mind	Google (Alphabet Inc.)	Sundar Pichai

Activity 1.5

Visit different AI image-generating tools such as **DALL·E, Craiyon,** or any other popular image chatbot. Try creating fun or educational images using prompts like "a robot teacher in a classroom" or "students learning with AI". Choose your favorite image and display it with a short caption describing the scene. *Example:* "A robot teaching students about space in a smart classroom."

5.5 Application of Integrated AI Tools

There are various AI tools which are integrated along with google docs, email and office 365. The integration of AI in these tools helps to organize the content and make work easier. Some of the most common features that AI provides are grammar corrections, voice typing, smart compose, etc.

Application of integrated AI tools in google docs:

i. Smart compose with auto correction
ii. Grammar and Spelling
iii. Voice typing
v. Language translation
vi. Auto summary

Application of integrated AI tools in email:

i. Smart compose and quick reply
ii. Schedule email
iii. Spam filtering
iv. Grammar and writing suggestions
v. Email summarization

Application of integrated AI tools in Office 365

i. Word AI powered writing tools
ii. AI driven data analysis in excel
iii. AI enhanced presentation
iv. AI powered meetings and collaboration

5.6 Ethics in AI (Bias, Privacy, and Security)

Ethics is the set of moral principles that defines what is wrong and what is right. Ethics guides how people should behave and decide what is good or bad, fair or unfair. Ethics is also essential in AI for making AI machines fair, safe and responsible. It ensures that AI can perform work properly without causing harm, respecting privacy, fairness and transparency.

Bias in AI: Bias means making unfair decisions or treating people differently based on things like gender, race, or background. AI systems should be fair and equal to everyone. However, if the data used to train an AI is unfair or incomplete, the AI may also behave unfairly. For example, if a hiring AI is trained only on resumes from men, it might ignore women's applications. This is not fair. Since AI learns from past data, it can repeat human biases unless carefully checked and corrected.

Privacy in AI: Privacy refers to the process of keeping the personal information and other details secret and safe from unauthorized access. AI collects various personal information and stores them and use them. The collected data should be kept secret because if data is misused it may lead to identity theft, spying or unfair treatment.

Security in AI: AI security means protecting AI systems from threats like hacking or data tampering. Since AI learns from training data, if the data is changed or attacked, the AI can make wrong decisions. For example, a hacked self-driving car might not stop at a red light. To keep AI safe, strong security methods like encryption and threat detection are used.

5.7 Internet of Things (IoT) and its Applications

Internet of Things (IoT) is the network of physical objects such as vehicles, TV, etc. which collects data through sensors and exchanges using the internet without human intervention. IoT makes life easier by automating tasks, saving energy, and helping emergencies. For example, the smoke detectors alert fire fighters as they detect smoke.

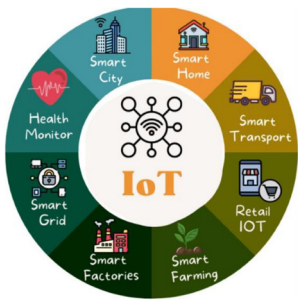


Fig 5.8

Application of IoT

i. Smart education
ii. Smart home
iii. Smart city
iv. Smart transportation
v. Smart healthcare
vi. Smart agriculture
vii. Communication

5.8 Virtual and Extended Reality (XR)

Virtual Reality (VR) is a computer-made environment that feels real. With tools like VR headsets, people can explore virtual worlds, play games, or even join virtual classrooms.

Extended Reality (XR) is a wider term that includes all technologies that mix the real and digital worlds. This includes:

- **VR** – full virtual worlds

- **AR (Augmented Reality)** – where digital images or info appear on top of the real world using a phone or special glasses

These technologies make learning, gaming, and exploring more fun and interactive.

Applications of VR and XR:

a. Education
b. Training
c. Health and medication
d. Business
e. Entertainment
f. Military and defense
g. Tourism and hospitality
h. Aeronautic

5.9 Cloud Computing and Their Applications

Cloud computing is a technology that allows people to use services like data storage, servers, software, and databases through the internet. Instead of having everything installed on your computer, you can access these services online from anywhere.

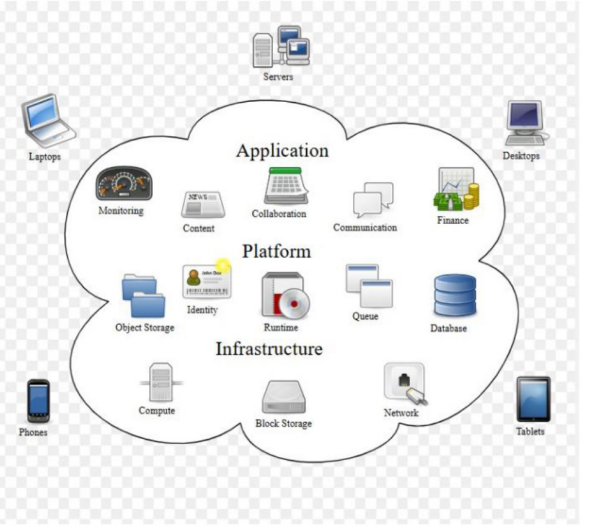


Figure 5.9

Companies like **Amazon Web Services (AWS)**, **Microsoft Azure**, **Google Cloud**, and **IBM Cloud** provide cloud services. Earlier, individuals or organizations had to buy and maintain their own hardware and software. But with cloud computing, everything is available online, no maintenance needed.

Examples of cloud-based apps include **Google Drive**, **Dropbox**, **OneDrive**, and **iCloud**. These tools let you store documents, photos, and videos online and access them anytime, anywhere without using pen drives or hard disks.

Benefits of cloud computing:

i. Improves teamwork and productivity
ii. Reduces hardware costs
iii. Keeps data safe and always available
iv. Provides access to advanced tools and services

Features of cloud computing:

i. On-demand service
ii. Access anytime, anywhere
iii. Scalable resources (can grow as needed)
iv. Shared resources (resource pooling)
v. Strong security
vi. Easy to maintain
vii. Low cost
viii. Reliable performance

Do you know?

- Smart cities use AI and IoT to manage traffic, save electricity, and even predict garbage collection needs.
- Cloud computing allows you to store and access your files from anywhere even from the top of a mountain if there's internet!
- AI helps detect cyber threats and protects your online data in real-time.

Services of cloud computing:

a. **SaaS (Software as a Service):** You can use various software applications without installing in your local device through the use of the internet. This

feature is known as software as a service. **Gmail, Google docs, office 365** etc. are some of the most commonly used software through cloud computing.

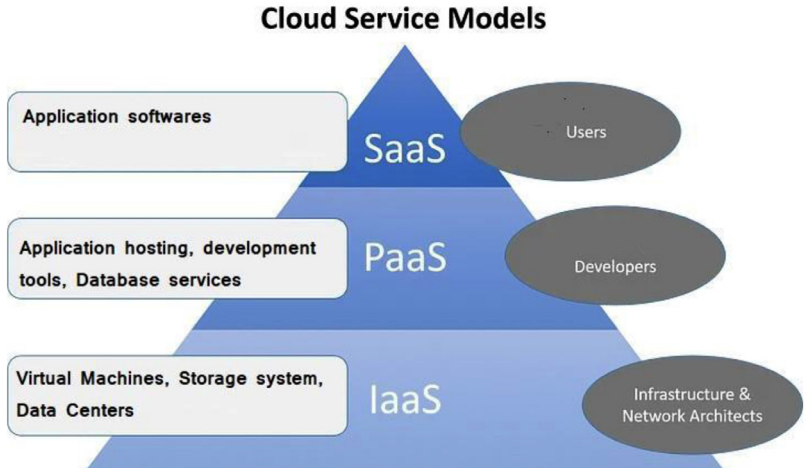


Figure 5.10

b. **PaaS (Platform as a Service):** Platform is the environment that helps to develop, build, test and run the software application. Cloud platform includes various tools, programming languages and data you can develop various software applications. Eg. **Google app engine, Microsoft azure app** services are examples of PaaS.

c. **IaaS (Infrastructure as a Service):** Cloud computing provides various hardware resources such as servers, storage and networks over the internet. The main purpose of IaaS is to help users avoid the cost and complexity of purchasing and managing the physical servers and hardware. For example, you can use **google drive, OneDrive** for storing your documents.

Activity 1.6:

Study the technology behind cloud storage (like Google Drive, OneDrive), its importance and how it is applied in real life scenarios. Explore how it works, and prepare a simple presentation and present in your class.

5.10 Concept of e-Commerce, e-Governance and e-Education

The development of ICT tools and the internet has changed how people do their daily activities. Today, people can run businesses, shop online, and get government services from their homes. There is no need to wait in long queues at offices.

Figure 5.11

e-commerce means buying and selling goods or services through the internet. It is also called **electronic commerce** and is becoming more popular around the world.

Merits of e-commerce

i. You can shop from your home.
ii. It saves time.
iii. Goods and services are available 24/7.
iv. You can access international products.
v. Items are delivered to your doorstep.

Limitations of e-commerce

i. The product delivered may be different from what was shown.
ii. There is a risk of online scams or fraud.
iii. It reduces face-to-face communication and social interaction.
iv. It increases dependency on technology.
v. Sometimes, delivery of goods may be delayed or not on time.

e-Governance

With the growing use of modern digital tools in daily life, governments around the world have also adopted technology to improve their services. e-Governance refers to the use of digital devices, the internet, and communication technologies to provide government services and information to citizens.

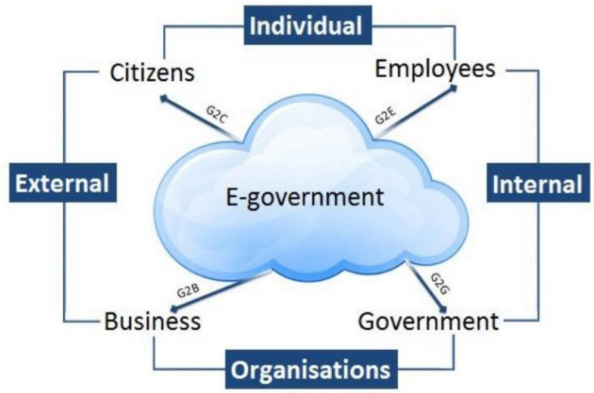


Figure 5.12 Types of e-Governance Classification

Through e-Governance, people can now access services like applying for passports, licenses, citizenship certificates, and more right from their homes using government websites. It is a shift from traditional manual systems to a digital platform, aiming to make governance more efficient, transparent, and citizen-friendly.

Merits of e-governance

i. Improves transparency - Citizens can track services and processes more openly.

ii. Ensures accountability - Officials become more responsible for their duties.

iii. Faster service delivery - Reduces time and paperwork.

iv. Equal accessibility - Makes services available to people in all regions.

v. Reduces corruption - Minimizes the need for in-person interactions, lowering chances of bribery.

Limitations of e-Governance

i. Digital divide - Not everyone has equal access to technology and the internet.

ii. Technical failures - System crashes or errors can interrupt service.

iii. Lack of digital literacy - Many people still lack the skills to use e-services effectively.

e-Education

Figure 5.13

In the past, education mainly took place in schools, with books as the primary source of knowledge. Students had to be physically present in the classroom to learn, and those who were absent missed the lessons. However, in the 21st century, learning is no longer limited to the classroom, teachers, or textbooks. With the rise of the internet and digital tools, students now have access to various online learning platforms that offer lessons, tutorials, and even complete degree programs. Some universities even conduct classes and exams entirely online.

e-Education refers to the use of modern digital technologies and the internet to carry out teaching and learning activities. It allows learners to access lessons, videos, assignments, and exams through devices like computers, tablets, or smartphones from anywhere and at any time.

Merits of e-Education

i. Learners can study at their own pace.

ii. It offers access to a wide variety of learning materials.

iii. Learning can happen anytime and anywhere.

iv. It is often more affordable than traditional education.

v. It provides convenience and flexibility.

Limitations of e-Education

i. Lack of face-to-face interaction between teachers and students

ii. Technical issues may affect the learning experience.

iii. The quality of content and instruction can vary.

Activity 1.7:

Create a list of emerging technologies that are most popular in today's era of IT. Mention their key features and how they have influenced the IT industry.

Exercise

1. Choose the correct answer.

i. Which of the following is not related to AI?

a. Behaving like human

b. Learning from past experience

c. Putting human brain in machine

d. Solving problems themselves.

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

a. Self driving cars

b. Siri

c. Alexa

d. All of the above.

iii. Which of the following is not an application of AI?

a. Gaming

b. Automated grading

c. Expert system

d. Manual typewriting

iv. In which type of learning is trained data used with correct answers?

a. Supervised

b. Supervised

c. Unsupervised

d. Semi unsupervised

v. Which of the following learning has no hints or labels in data?

a. Supervised

b. Supervised

c. Unsupervised

d. Semi unsupervised

vi. **Choose the AI chatbot developed by Google.**

a. Deepseek
c. ChatGPT
b. Copilot
d. Gemini

vii. **If you are a programmer, which chatbot will be more beneficial for you?**

a. Copilot
c. Deepseek
b. ChatGPT
d. Gemini

viii. **Which of the following is a service of cloud computing?**

a. IaaS
c. PaaS
b. Saas
d. All of the above

ix. **What is the process of buying and selling goods online called?**

a. E-commerce
c. Online shopping
b. Daraz
d. None of the above

x. **Which of the following is an e-Education tool?**

a. Google classroom
c. ChatGPT
b. Zoom
d. Google meet

xi. **Which of the following is an example of e-governance?**

a. Online driving license renewal
b. Online passport
b. Online fuel
d. Online result

2. Write short answers to these questions.

a. Write any two popular e-commerce sites in Nepal.
b. Define e-governance.
c. Mention any two applications of AI.
d. Which chatbot is useful for generating texts and images? Mention any two.
e. List any two applications of VR.
f. Write two advantages of e-education.

g. Write two examples of e-governance in Nepal.

h. What are the two popular e-commerce websites of Nepal?

i. Write two applications of cloud computing.

j. List any four popular generative AI.

3. Write long answers to these questions.

a. Define Artificial Intelligence (AI). Describe at least five of its practical applications in daily life.

b. What is machine learning (ML)? Explain how machine learning is related to Artificial Intelligence with suitable examples.

c. Describe any four major applications of machine learning in real-world scenarios such as education, healthcare, or transportation.

d. Differentiate between supervised and unsupervised learning. Use examples to illustrate how each type functions.

e. Explain the role of robotics in modern industries. Discuss how robots have transformed various sectors like healthcare, manufacturing, or education.

f. What is Generative AI? Discuss its importance and how it is changing the way we create digital content such as images, videos, and text.

g. Discuss the role of the Internet of Things (IoT) in the development of smart cities. Include examples of how IoT improves urban life.

h. Analyze the merits and demerits of e-commerce. Provide examples to support your answer.

i. How does e-governance enhance service delivery to citizens? Discuss with examples from sectors like health, education, or transport.

j. Evaluate the advantages of e-education. How has it changed the traditional learning environment for students and teachers?

k. Define cloud computing. List and explain any four major services provided by cloud computing platforms.