Internet of Things
Software Development
IX

NAVTTC

National Vocational & Technical Training Commission (NAVTTC)

Textbook of

Internet of Things

Software Development

Grade – IX

National Vocational and Technical Training Commission

H-9, Islamabad

Author: **Muhammad Umair** researcher,Sensors Cloud and Services (SCS) Lab, School of Computer Sciences

Reviewers:

1. Dr. Ahmed Mustafa, Chief Instructor, P-TEVTA, Lahore
2. Dr. Hina Khalid, Assistant Professor, UET, Lahore
3. Mr. Atif Anees, DACUM Facilitator
4. Ms. Mansoorah Ibrahim, Education Officer, NCC, Islamabad
5. Mr. Muhammad Asim, Coordinator, NAVTTC

Designing: Gul Awan Printers, Blue Area, Islamabad.

Edition: 1st Edition, 2022

ISBN:

Publishers: National Vocational & Technical Training Commission H-9, Islamabad.

Website: www.navttc.gov.pk,

Note: Names of Curriculum Developers will be part of curriculum document. In the book there will be names of reviewers only.

PREFACE

This book has been written so as to meet therequirements of the Matric Tech stream in the trade Internet of Things (IoT). This book is specific for IoT software development. Matric Tech in IoT has been introduced for the first time at SSC level. This textbook is the first national effort to describe the topics related to IoT software development in one book. A key attempt has been made to make the book interesting and useful. All the chapters presented cover the basic details required and understandable to the students of Matric Tech. All chapters includes assessments in form of MCQs, short questions and long questions.

The book covers 40% theory and 60% practical content. This content is equally helpful for the students of electronics, computer sciences and networking.

Any improvements and suggestions for the betterment of this book will be highly appreciated.

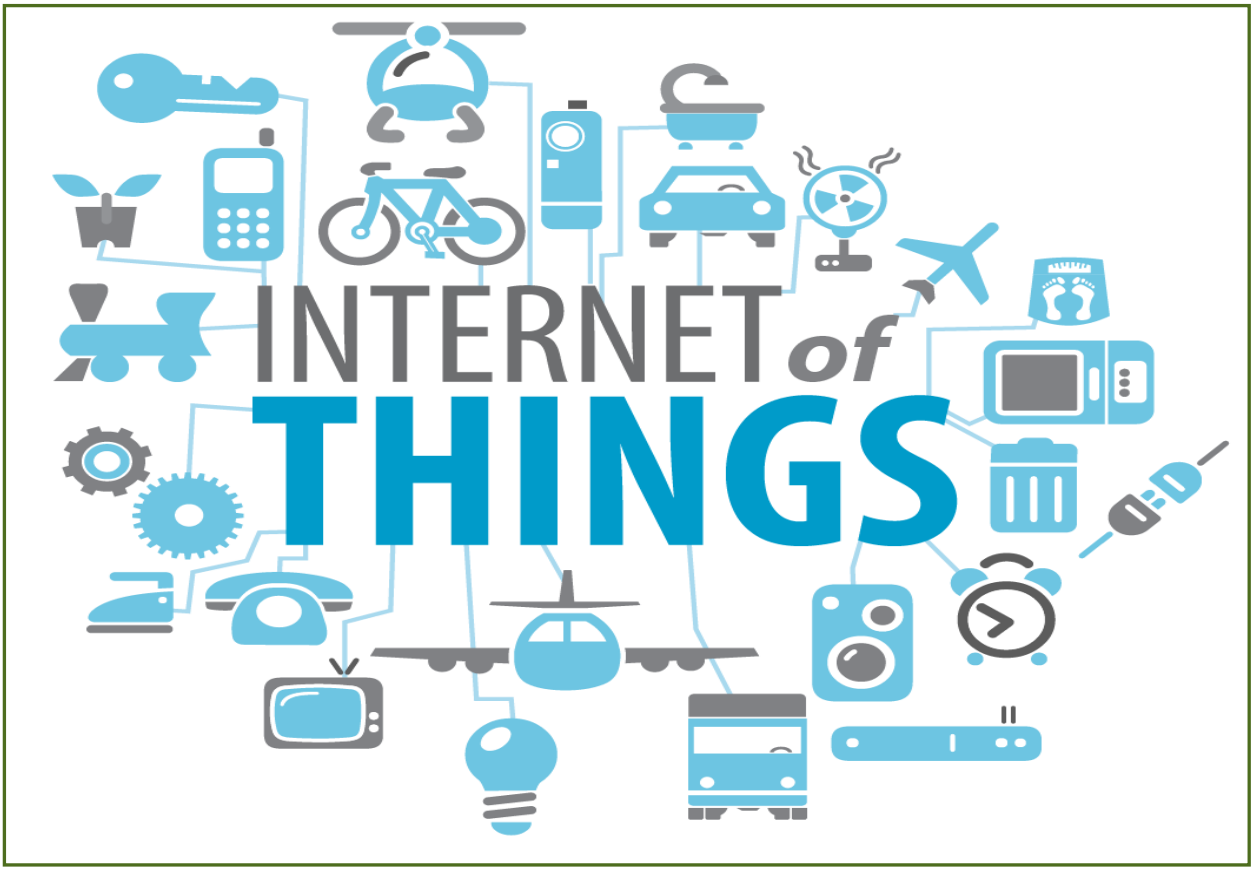
Executive Director
National Vocational & Technical Training Commission
(NAVTTC)

.

Table of Contents

Chapter 1

Introduction to Internet of Things



After Studying this chapter, you will be able to

- define IoT.
- know different components of IoT networking, messaging and hardware.
- understand the importance of IoT.
- evolution of IoT.
- know the scope of IoT in domestic, commercial, & industrial applications.
- learn different IoT applications.
- understand high level (basic) working of smart home system, remote monitoring system and smart cities.

1.1 Internet of Things

The Internet of Things (IoT) refers to the billions of physical devices around the world that are now connected to the internet, all collecting and sharing data. Before the advent of IoT, only the computers and mobile phones were capable of connecting to the internet. IoT enables connection of “things” with internet. “Things” can be sensors, actuators and devices containing a microcontroller. These different objects and sensors add a level of digital intelligence to devices that would be otherwise dumb, enabling them to communicate real-time data. The Internet of Things is making the fabric of the world around us smarter and more responsive, merging the digital and physical universes.

The term IoT is mainly used for devices that wouldn't usually be expected to have an internet connection, and that can communicate with the network independently of human action. For this reason, a PC isn't generally considered an IoT device and neither is a smartphone. However, a smartwatch or a fitness band or other wearable device might be counted as an IoT device.

> Do you know?
>
> Initial concept of IoT was to create a separate network of “things”. It was not necessary to connect things with the Internet.

1.1.1 Components of IoT

A typical IoT system consists of following main components:

Sensors/Devices

Sensors or devices help in collecting very minute data from the surrounding environment. All of this collected data can have various degrees of complexities ranging from a simple temperature monitoring sensor or a complex full video feed.A device can have multiple sensors. For example, our phone is a device that has multiple sensors such as GPS, accelerometer, camera but our phone does not simply sense things.

Connectivity

The sensors can be connected to the cloud through various mediums of communication such as cellular networks, satellite networks, Wi-Fi, Bluetooth, wide-area networks (WAN), low power wide area network and many more.Each option has some specifications and trade-offs between power consumption, range, and bandwidth. So, choosing the best connectivity option in the IOT system is important.

Data Processing

Once the data is collected and it gets to the cloud, the software performs processing on the acquired data.This can range from something very simple, such as checking that the temperature reading on devices such as AC or heaters is within an acceptable range. It can sometimes also be very complex, such as identifying objects (such as intruders in your house) using computer vision on video.There might be a situation when a user interaction is required.

User Interface

In an IoT systems, the information is made available to the end-user in some way. This can be achieved by triggering alarms on their phones or notifying through texts or emails.A usermight also have an interface through which they can actively check in on their IoT system. For example, a user has a camera installed in his house, he might want to check the video recordings and all the feeds through a web server.However, it’s not always this easy and a one-way street. Depending on the IoT application and complexity of the system, the user may also be able to perform an action that may backfire and affect the system.For example, if a user detects some changes in the refrigerator, the user can remotely adjust the temperature via their phone.There are also cases where some actions take place automatically. By establishing and implementing some predefined rules, the entire IoT system can adjust the settings automatically and no human has to be physically present.In case any intruders are sensed, the system can generate an alert not only to the owner of the house but to the concerned authorities. Figure 1.1 shows user dashboard of an IoT enabled smart home.

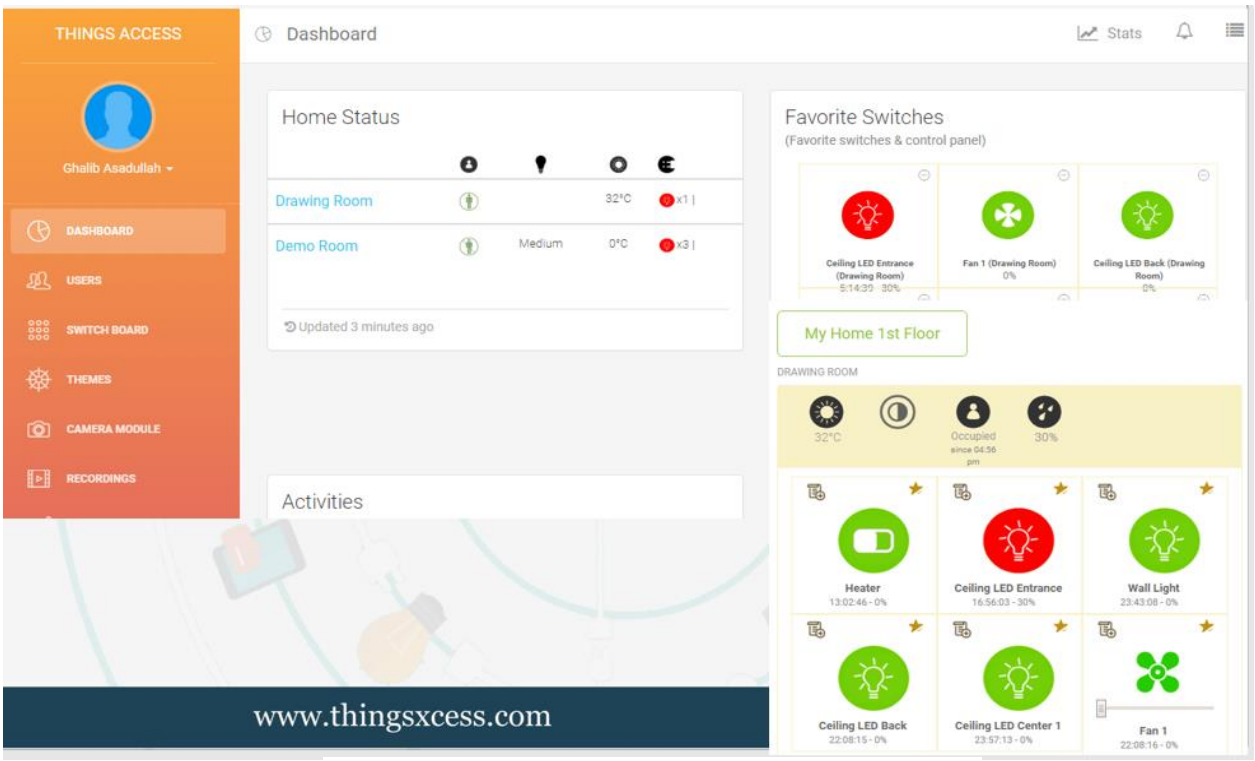


Fig. 1.1 User Interface of a Smart Home

Teacher Notes:

- Give real-life live examples of IoT systems to students.

Figure 1.2 shows an example of an IoT system.

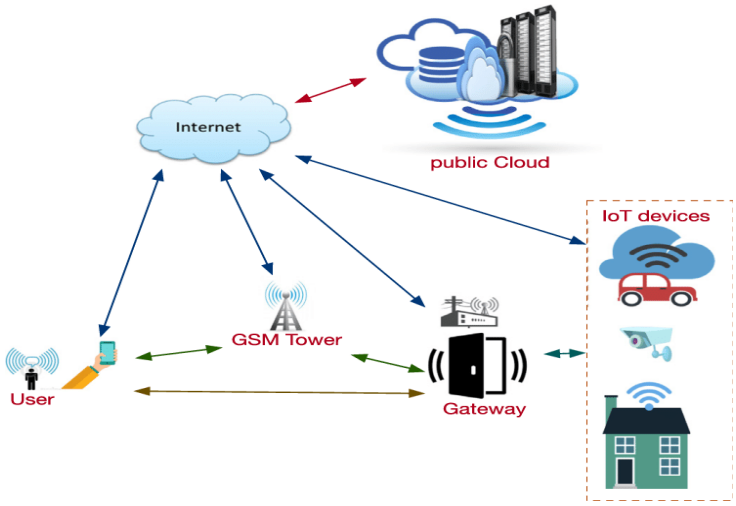


Fig. 1.2 Example of an IoT System

1.1.2 Importance of IoT

Consider an IoT use case: you went to the market and forgot to switch off your AC, fan, or light and felt helpless that you cannot return home to switch it off? This is where IoT comes into the picture. It can remind you of the essential tasks that you often forget to do. Also, using the mobile application, you can access your home appliances integrated with IoT from anywhere around the world.

Another very exciting IoT device is the NFC (near field communication) smart ring.These rings serve as a multi-purpose device. They are rings consisting of the connectivity of the network, NFC chips, and sensors that help you exchange data. By using NFC rings, you can pay your bills, access your car's door lock by just swiping it, and also get your mobile notifications. ANFC ring is shown in Figure 1.3.

Fig. 1.3 Example of NFC Ring

All this is possible due to the connection of the IoT devices over the internet servers that helps in sharing and exchanging the data. This technology reduces human efforts and saves a lot of time.

1.2 Scope of IoT

Internet of Things has emerged as a leading technology around the world. It has gained a lot of popularity in lesser time. Also, the advancements in Artificial Intelligence and Machine Learning have made the automation of IoT devices easy. Basically, AI and ML programs are combined with IoT devices to make them intelligent. Due to this, IoT has also expanded its area of application in various sectors. IoT applications are found in smart homes, smart buildings, smart cities, smart agriculture and smart transportation etc. Figure 1.4 enlists scope of IoT in different domains.

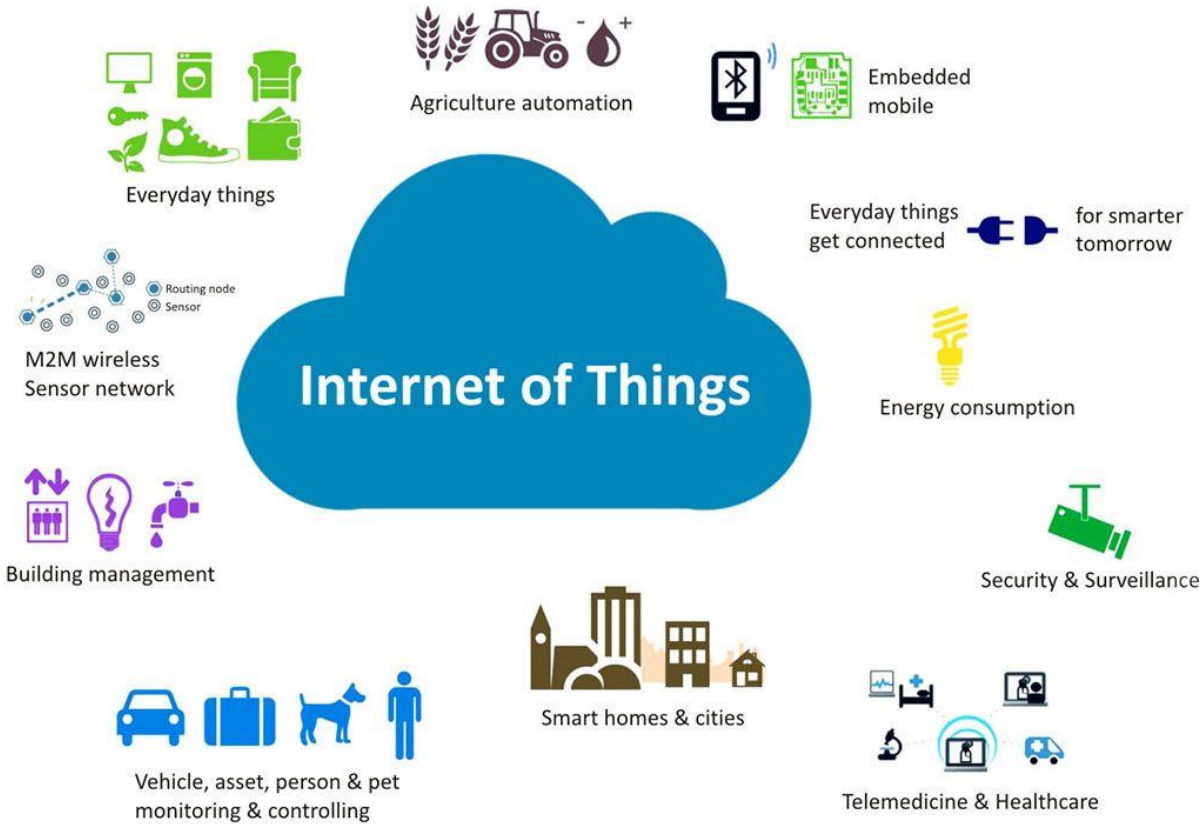


Fig. 1.4 Scope of IoT

Do you know?

There are expected to be more than 30.9B IoT devices worldwide by 2025. According to the latest research, the number of IoT-connected devices globally reached 11.7 billion in 2020.

1.3 Verticals of IoT

The following are some IoT vertical markets that have most noticeably adopted IoT solutions to date.

1.3.1 Smart Buildings

Smart building management involves collecting data from smart devices and sensors to remotely monitor a property's energy, security, landscaping, HVAC (heating, ventilation and control), lighting and more. Actions can be automated according to events and efficiency can be optimized, saving time, resources and cost.Figure 1.5 shows components of a smart building.

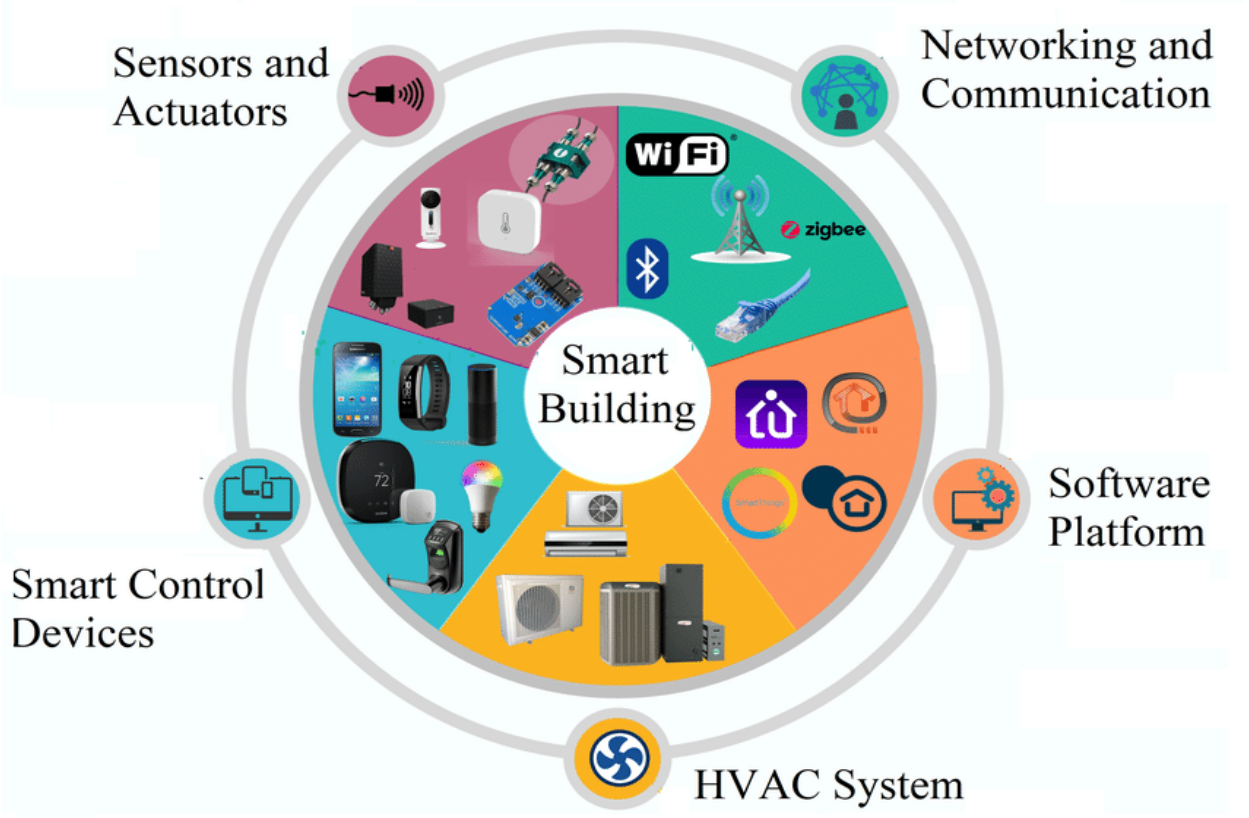


Fig. 1.5 Components of a Smart Building

1.3.2 Industrial IoT

Industrial IoT (IIoT) uses system integration and sensors to gather data within a process for analysing and optimizing different manufacturing and other processes. As a result, human error and operation cost is reduced. For example, as operation costs fluctuate in oil and gas sector, remote monitoring and insightful decision making can keep an enterprise in the oil andgas industry successful. Regulatory compliance can be accurately monitored and overall costs decrease.

Agriculture is one of the major industries constituting a large sector of economy. It involving food farming, livestock farming and cotton farming. With the ever-increasing population of the earth, it is important for the farming industry to operate as efficiently and effectively as possible. IoT can enable local and commercial farming to be more environmentally friendly, cost effective and production efficient. Figure 1.6 shows components of Industrial IoT.

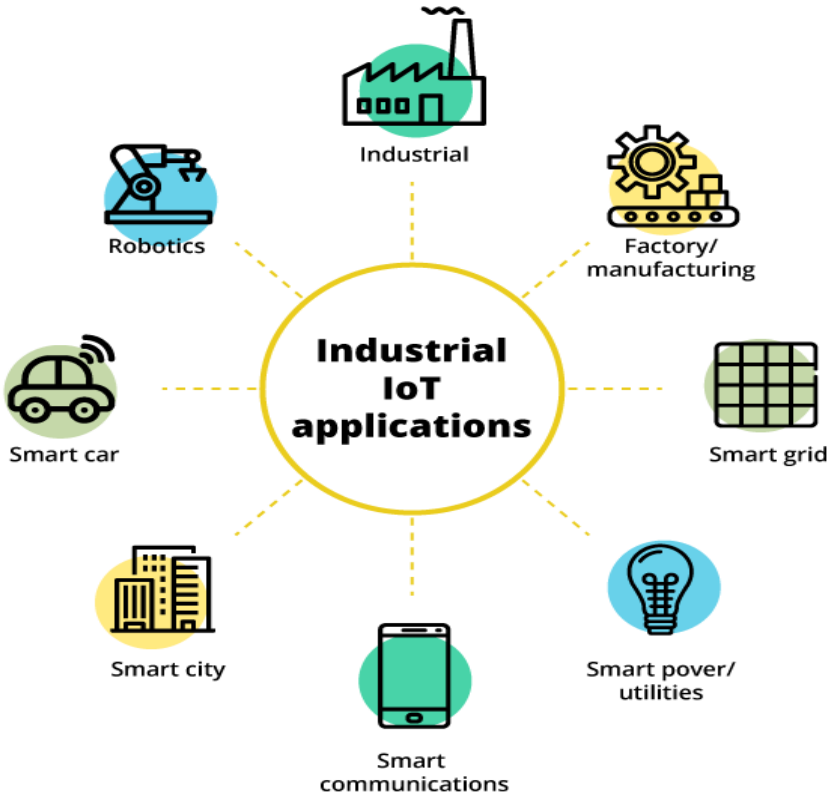


Fig. 1.6 Components of Industrial IoT

1.3.3 Smart Transportation and Logistics

IoT can also be employed in smart transportation and logistics management. In an IoT enabled smart transportation environment, fleet management, asset tracking and predictive maintenance work together as an end-to-end solution. Figure 1.7 shows components of a smart transportation system.

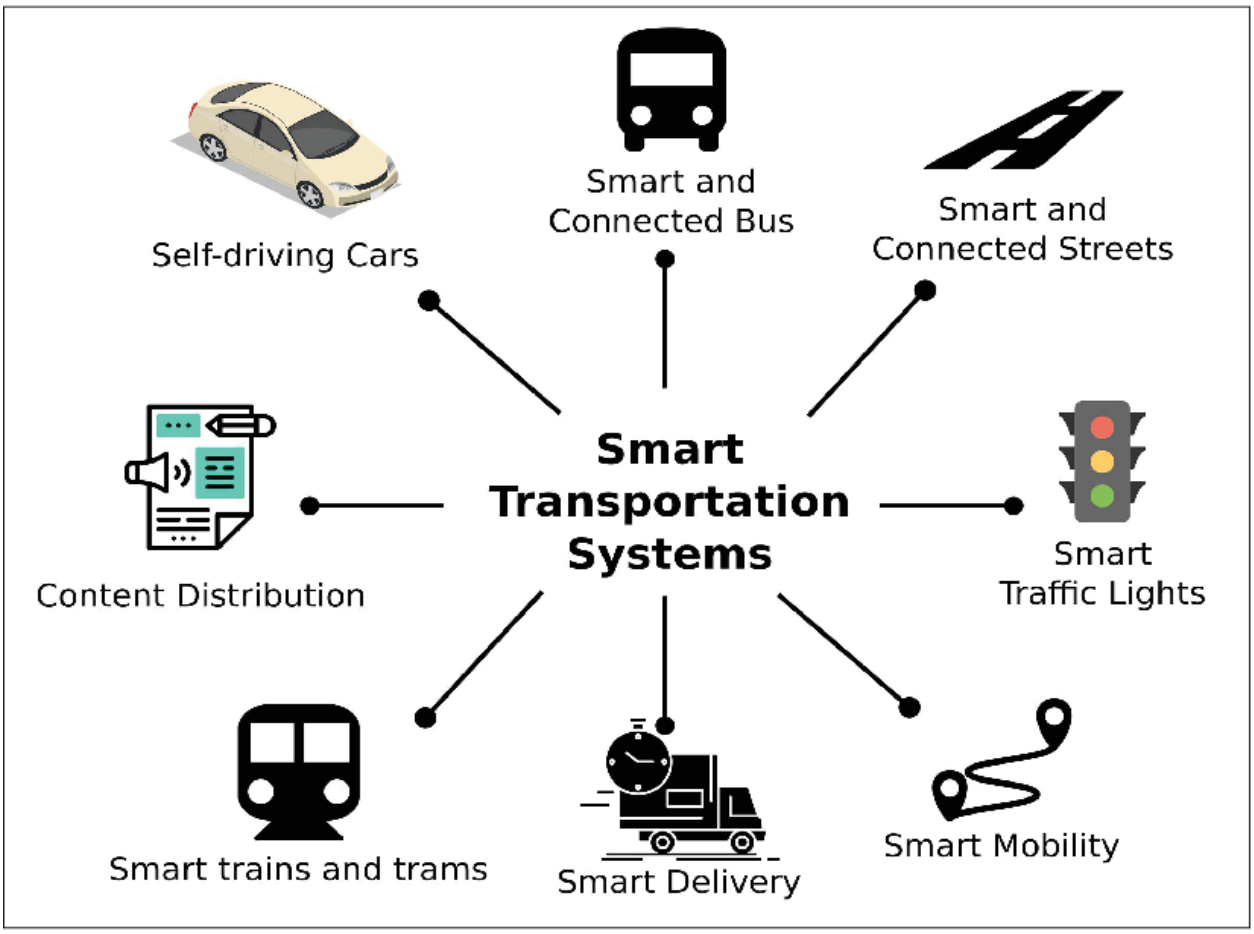


Fig. 1.7 Components of Smart Transportation

1.3.4 Smart Homes

Consumers connecting smart devices within homes can enhance the home experience, increase home security and conserve energy. Homeowners can now monitor their properties remotely and in real-time. Figure 1.8 shows components of a smart home.

Do you know?
The Google Smart Home platform lets users control your commercially available connected devices through the Google Home app and Google Assistant, which is available on more than 1 billion devices, like smart speakers, phones, cars, TVs, headphones, watches, and more.

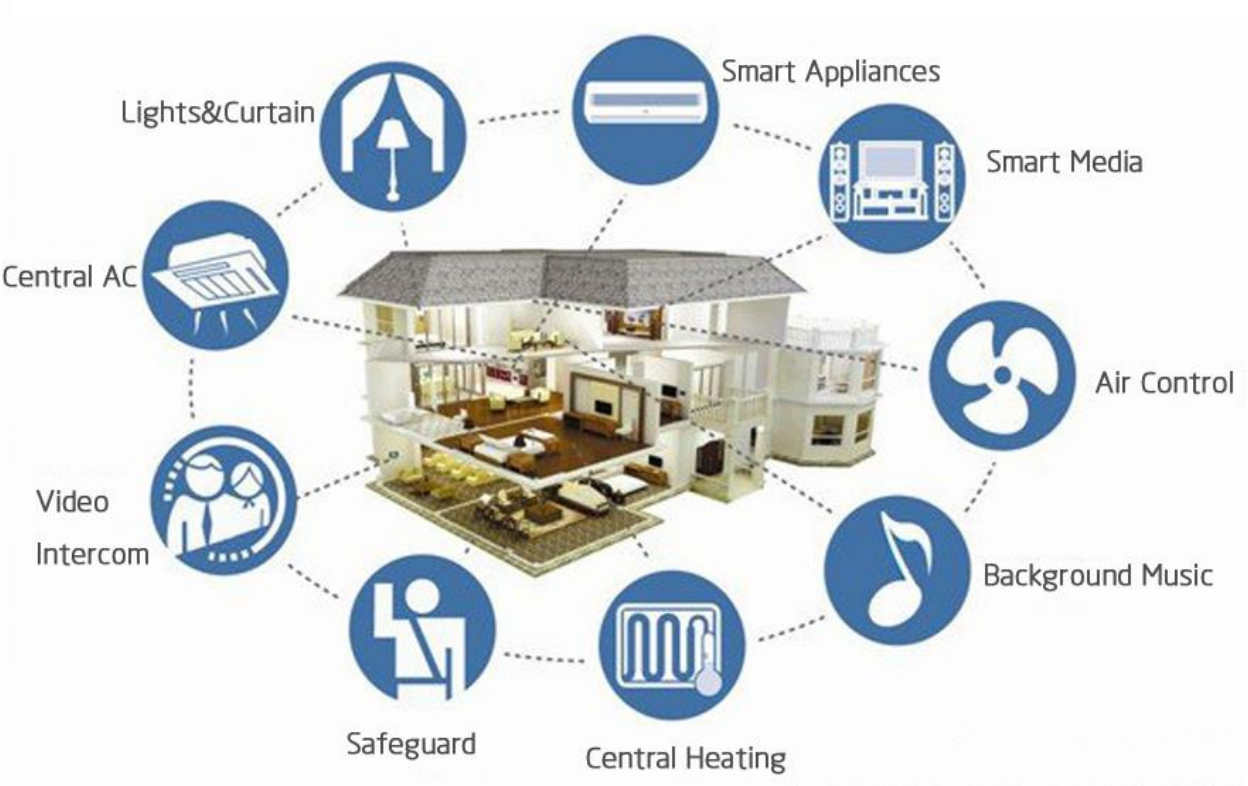


Fig. 1.8 Components of a Smart Home

1.3.5 Smart Cities

A smart city can include anything from smart parking to mass transit. A smart city addresses traffic, public safety, energy management and more for its government and citizens. More and more cities across the world are adopting these solutions at a steady rate. Figure 1.9 shows components of a smart city.

Interesting Information
- Singapore is the smartest city in the world, according to the IMD's inaugural Smart City Index.

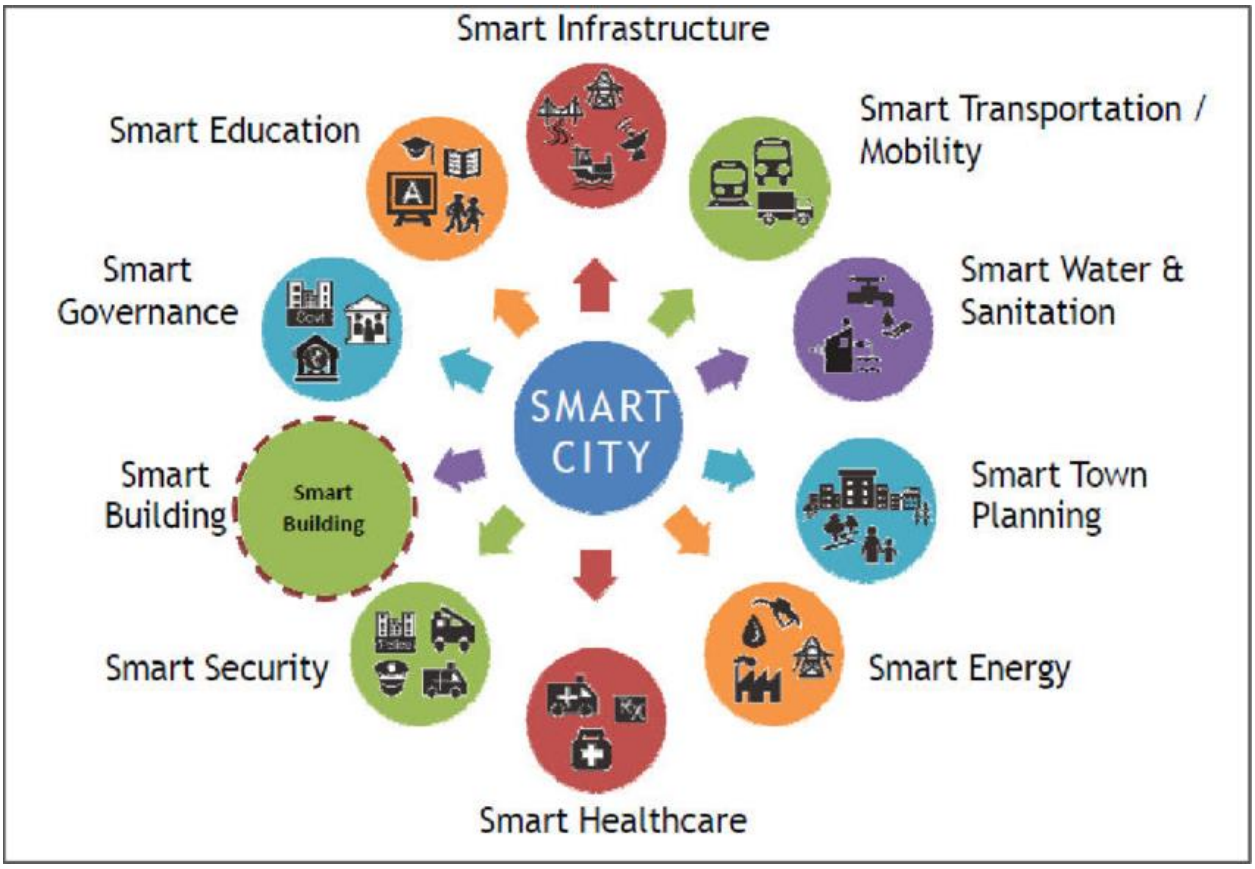


Fig. 1.9 Components of a Smart City

1.3.5 Smart Healthcare

Healthcare is perhaps one of the fastest adopters of smart, connected technology. IoT enables critical business and patient monitoring decisions to be made remotely and in real-time. Figure 1.10 shows components of a smart healthcare system.

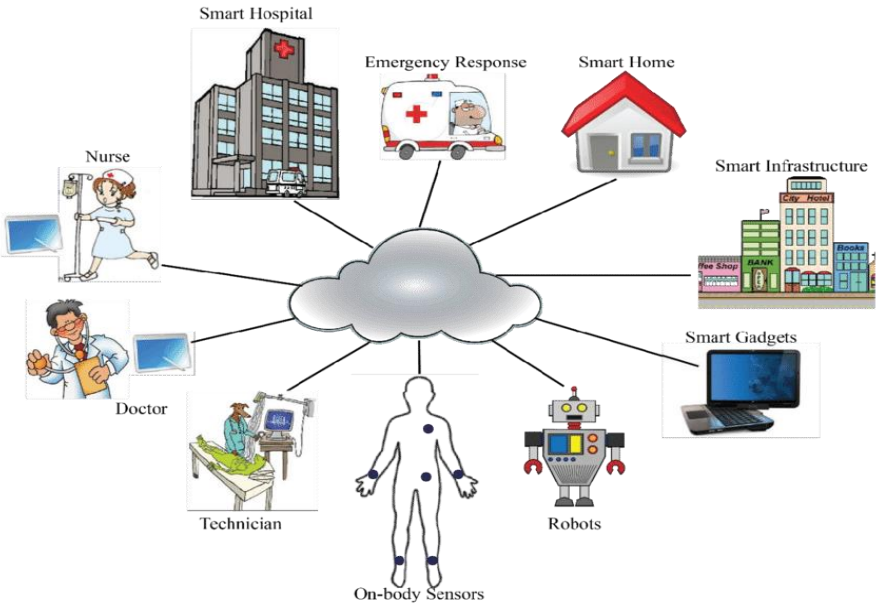


Fig. 1.10 Components of Smart Healthcare

These are just a few of the many vertical markets adopting Internet of Things solutions in the world today. Many of these solutions have no doubt had an effect on you, whether noticed or not. In fact, the seamless experience of these solutions is actually one of the most important aspects of IoT implementation and adoption by consumers. The acceleration of the internet of things will only be more prevalent moving forward.

Key Points

- The Internet of Things (IoT) refers to the billions of physical devices around the world that are connected to the Internet.
- An IoT system consists of Sensors, actuators, connectivity medium, IoT cloud and a user interface.
- IoT applications are found in smart homes, smart buildings, smart cities, smart agriculture and smart transportation etc.
- Smart building management involves collecting data from smart devices and sensors to remotely monitor a property's energy, security, landscaping, HVAC, lighting and more.
- Industrial IoT (IIoT) uses system integration and sensors to gather data within a process for analysing and optimizing different manufacturing and other processes.
- In an IoT enabled smart transportation environment, fleet management, asset tracking and predictive maintenance work together as an end-to-end solution.
- Consumers connecting smart devices within homes can enhance the home experience, increase home security and conserve energy.
- A smart city addresses traffic, public safety, energy management and more for its government and citizens.
- In healthcare, IoT enables critical business and patient monitoring decisions to be made remotely and in real-time.

Exercise

Select the most appropriate option

1. IoT stands for

   a. Internet of Devices b. Internet of triangles c. Internet of Trees d. Internet of Things

2. There will be around _____ billion IoT devices by 2025.

   a. 40 b. 20 c. 25 d. 30

3. Fleet managemnet lies in which IoT vertical?

   a. Smart Cities b. Smart Agriculture c. Industrial IoT d. Smart transportation

4. HVAC is a main component of which IoT vertical

   Smart Cities b. Smart Agriculture c. Industrial IoT d. Smart transportation

5. Which is the smartest city in the world

   a. Singaporeb. Helsinki c. Tokyo d. None of these

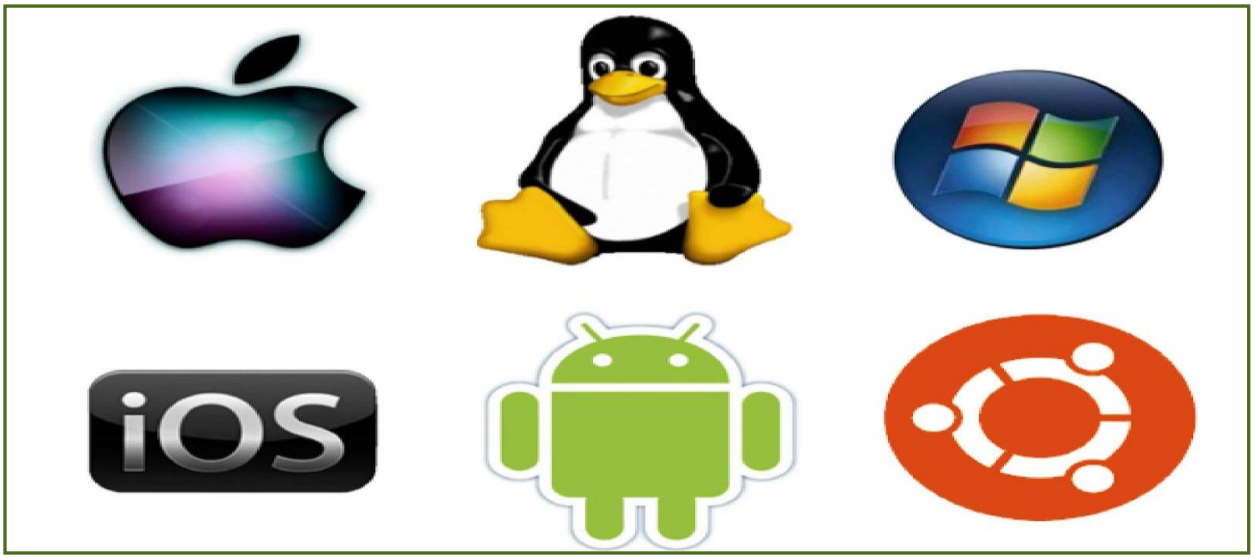
Write short answer of the following

1. Define IoT.
2. What is the importance of IoT?
3. What is the scope of IoT?
4. Enlist some applications of IoT.

Answer the following question in detai.

1. Explain different components of an IoT system.
2. Describe IoT verticals and their components.

Chapter 2
Operating Systems



After Studying this chapter, you will be able to

- define software and operating system.
- learn types of software: system software, application software.
- describe various types of operating systems: Windows, Linux.
- understand firmware.
- understand Installation of OS (windows and Linux).
- describe different steps of installation process.
- boot from storage media (DVD, Mass storage, external hard disk etc.).
- partitioning and formatting, selecting appropriate drive.
- install device driver.
- install operating system on virtual machine.
- know copying, moving, rename files and folders, search files and folders.
- personalize desktop settings and display settings.
- know files extensions, hide / unhide files / folders / system files.
- use device manager, task manager and command prompt.
- install application software on windows.
- differentiate between updating and upgrading.
- understand the process of updating.
- hard drive free space.
- downloading and applying updates to the OS.
- understand the process of upgrading.
- hard drive free space.
- booting from storage media (DVD, Mass storage, external hard disk etc.) or internet.
- selecting appropriate drive.

2.1System software v/s Application software

A software is a set of instructions or programs instructing a computer to do specific tasks. Software is basically a generic term used to describe computer programs. In general scripts, applications, programs and a set of instructions are terms often used to describe a software.

On the basis of language in which software is developed and platform which is required for its execution, we can classify software into two divisions: System software and Application software. The following are some basic differences between a System software and an Application software.

Key	**System Software**	**Application Software**
Definition	System software is an interface between application software and system.	Application software is the type of software which runs as per user request. It runs on a platform which is provided by the system software.
Development Language	In general, system software are developed in low level language which is more compatible with the system hardware in order to interact with.	While in case of Application software high level language is used for their development as they are developed as some specific purpose software.
Usage	System software is used for operating computer hardware.	Application software is used by user to perform specific tasks.
Installation	System software are installed on the computer when operating system is installed.	Application software are installed according to user's requirements.
User interaction	System software are specific to system hardware. Therefore, very less or no user interaction available in system software.	Application software user can interact with it using a user interface.
Dependency	System software can run independently. It provides platform for running application software.	Application software can't run independently. They can't run without the presence of a system software.
Examples	Some examples of system software's are compiler, assembler, debugger, driver, etc.	On other hand some examples of application software's are word processor, web browser, media player, etc.

2.2 Operating System Abstractions

An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers. Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS etc.

Definition

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

2.2.1 Functions of an Operating System

An operating system performs various functions as shown in Figure 2.1. The following are some of important functions of an operating System.

- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users

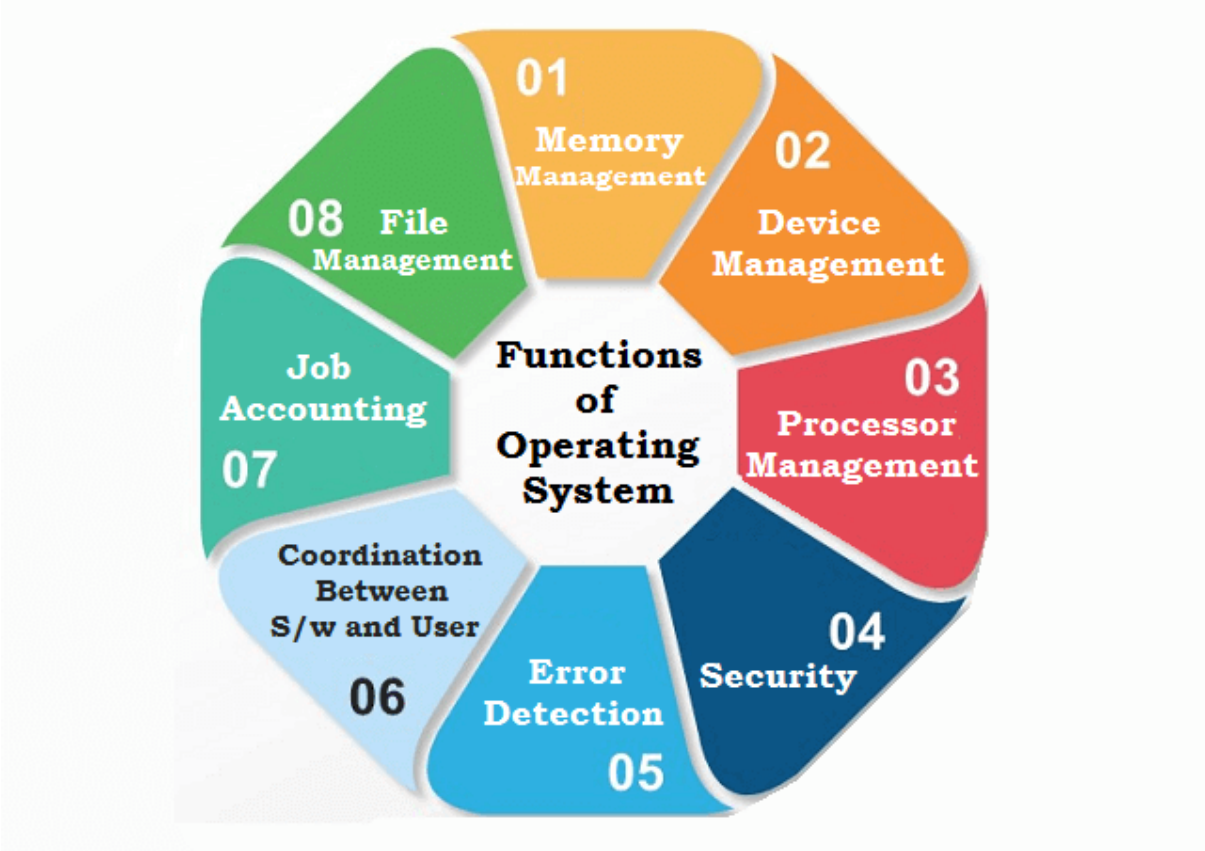


Fig. 2.1 Functions of an Operating System

Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words (set of 2 bytes) or bytes where each word or byte has its own address.Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. An Operating System does the following activities for memory management:

- Keeps tracks of primary memory, i.e., which parts of it are in use by whom, which parts are not in use.
- In multiprogramming, the OS decides which process will get memory when and how much.
- Allocates the memory when a process requests it to do so.

- De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called **process scheduling**. An Operating System does the following activities for processor management:

- Keeps tracks of processor and status of processes. The program responsible for this task is known as **traffic controller**.
- Allocates the processor (CPU) to a process.
- De-allocates processor when a process is no longer required.

Device Management

An Operating System manages device communication using respective drivers. It does the following activities for device management:

- Keeps tracks of all devices. Program responsible for this task is known as the I/O controller.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions. An Operating System does the following activities for file management:

- Keeps track of information, location, uses and status etc.
- Decides who gets the resources.

- Allocates the resources.
- De-allocates the resources.

Other Important Activities

Following are some of the important activities that an Operating System performs:

- **Security**: By means of password and similar other techniques, it prevents unauthorized access to programs and data.
- **Control over system performance**: Recording delays between request for a service and response from the system.
- **Job accounting**: Keeping track of time and resources used by various jobs and users.
- **Error detecting aids**: Production of dumps, traces, error messages, and other debugging and error detecting aids.
- **Coordination between other software and users**: Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

2.2.2 Examples of Operating Systems

Linux and Windows both Operating systems have their advantages and disadvantages. Windows is simple to use but is not a free and open-source OS, whereas Linux is free, open source, customizable and secure but kind of complex for the users having no programming background. Linux is more reliable than windows. The prior difference between Linux and Windows operating system is that Linux is totally free of cost whereas windows is marketable operating system and is costly. An operating system is a program intended to control the computer hardware and behave as an intermediary between user and hardware.

2.3 Firmware

In electronic systems and computing, **firmware** is a tangible electronic component with embedded software instructions, such as a BIOS. Typically, those software instructions are used to tell an electronic device how to operate. Typical examples of devices containing firmware are embedded systems (such as traffic lights, consumer appliances, and digital watches), computers, computer peripherals, mobile phones, and digital cameras. The firmware contained in these devices provides the control program for the device.

Firmware is stored in non-volatile memory devices such as ROM, EPROM, or flash memory. Changing the firmware of a device may rarely or never be done during its economic lifetime; some firmware memory devices are permanently installed and cannot be changed after manufacture. Common reasons for updating firmware include fixing bugs or adding features to the device. This may require ROM integrated circuits to be physically replaced, or flash memory to be reprogrammed through a special procedure. Firmware such as the ROM BIOS of a personal computer may contain only elementary basic functions of a device and may only provide services to higher-level software. Firmware such as the program of an embedded system may be the only program that will run on the system and provide all of its functions.

Interesting Information

BIOS, computing, stands for Basic Input/ Output System. The BIOS is a computer program embedded on a chip on a computer's motherboard that recognizes and controls various devices that make up the computer.

Do you know?

The first operating system was created by General Motors in 1956 to run a single IBM central computer. In the 1960s, IBM was the first computer manufacturer to take on the task of developing operating systems and began distributing operating systems included in its computers.

2.4 Installing Ubuntu

The Ubuntu desktop is easy to use, easy to install and includes everything you need to run your organization, school, home or enterprise. It's also open source, secure, accessible and free to download.

It is easy to install Ubuntu from a DVD. Follow the following steps to install Ubuntu:

1. Put the Ubuntu DVD into your optical/DVD drive.
2. Restart your computer.

As soon as your computer boots, you'll see the welcome window (Figure 2.2).

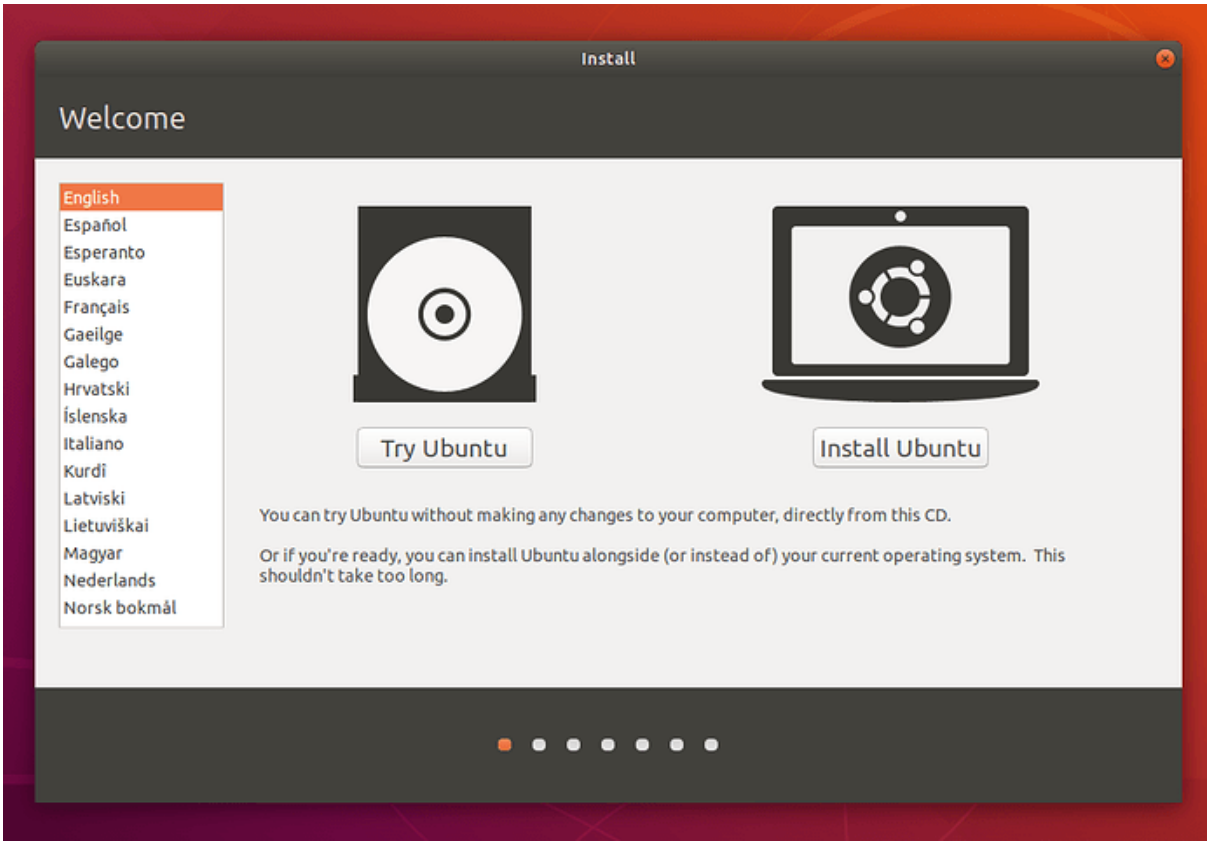


Fig. 2.2 Ubuntu Installation: Welcome Window

Select 'Install Ubuntu', and press return to launch the desktop installer automatically. Alternatively, select the first option, 'Try Ubuntu without installing', to test Ubuntu.

Boot from USB Flash Drive

Most computer sboot from USB automatically. Simply insert the USB flash drive and turn on your computer. You should see the same welcome window we saw in the previous 'Install from DVD' step, prompting you to choose your language and install or try the Ubuntu desktop. If your computer doesn't automatically boot from USB, try holding F12 when your computer starts. This will allow you to select the USB device from a system-specific boot menu. F12 is the most common key for bringing up your system's boot menu, but Escape+F2 and F10 are common alternatives.

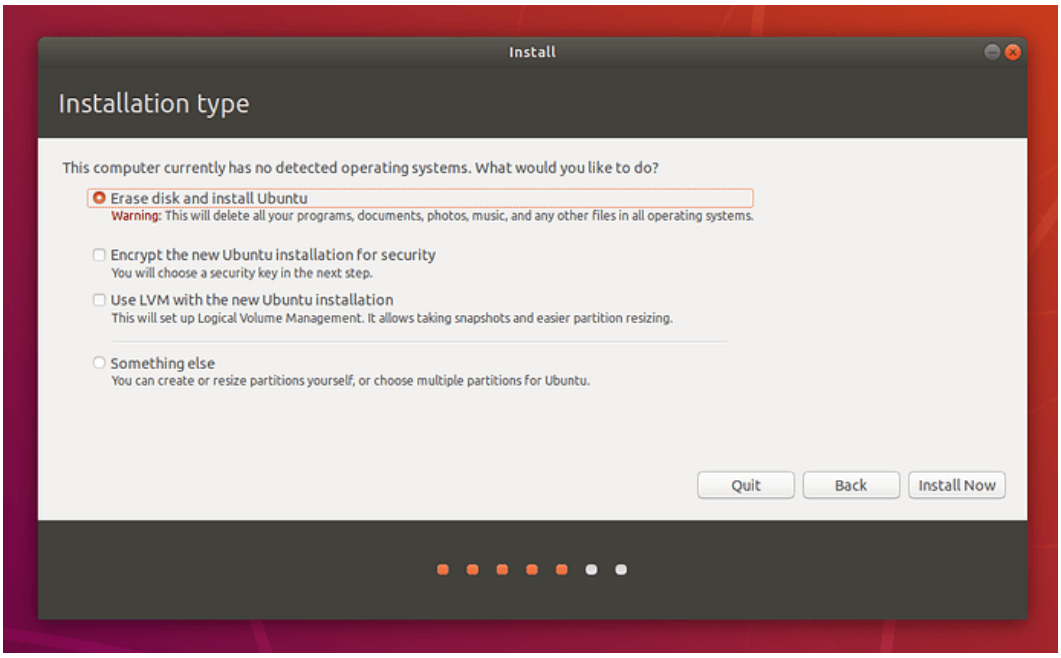
Prepare to install Ubuntu

After booting from selected storage media, installation wizard asks you to select your keyboard layout. If the installer doesn't guess the default layout correctly, use the 'Detect Keyboard Layout' button to run through a brief configuration procedure. After selecting '*Continue'* you will be asked '*What apps would you like to install to start with?'* The two options are 'Normal installation' and 'Minimal installation'. The first is the equivalent to the old default bundle of utilities, applications, games and media players. The second takes considerably less storage space and allows you to install only the required components.

Below the "installation-type" question are two checkboxes; one to enable updates while installing and another to enable third-party software.

Allocate Drive Space

Use the checkboxes to choose whether you'd like to install Ubuntu alongside another operating system, delete your existing operating system and replace it with Ubuntu, or — if you're an advanced user — choose the 'Something else' option (Figure 2.3).



Do you know?

The word "Ubuntu" is an ancient Zulu and Xhosa word that means "humanity to others". Ubuntu also means "I am what I am because of who we all are". It was chosen because these sentiments precisely describe the spirit of the Ubuntu distribution.

Begin Installation

After configuring storage, click on the '*Install Now'* button. A small pane will appear with an overview of the storage options you've chosen, with the chance to go back if the details are incorrect. Click Continue to apply those changes in place and start the installation process.

Select Location

If you are connected to the internet, your location will be detected automatically. Check your location is correct and click 'Forward' to proceed. If you're unsure of your time zone, type the name of a local town or city or use the map to select your location.

Login Details

In the next step, a login window (Figure 2.4) will appear. Enter your name and the installer will automatically suggest a computer name and username. Username will be

your login and account name. Next, enter a strong password. You can also choose to enable automatic login and home folder encryption.

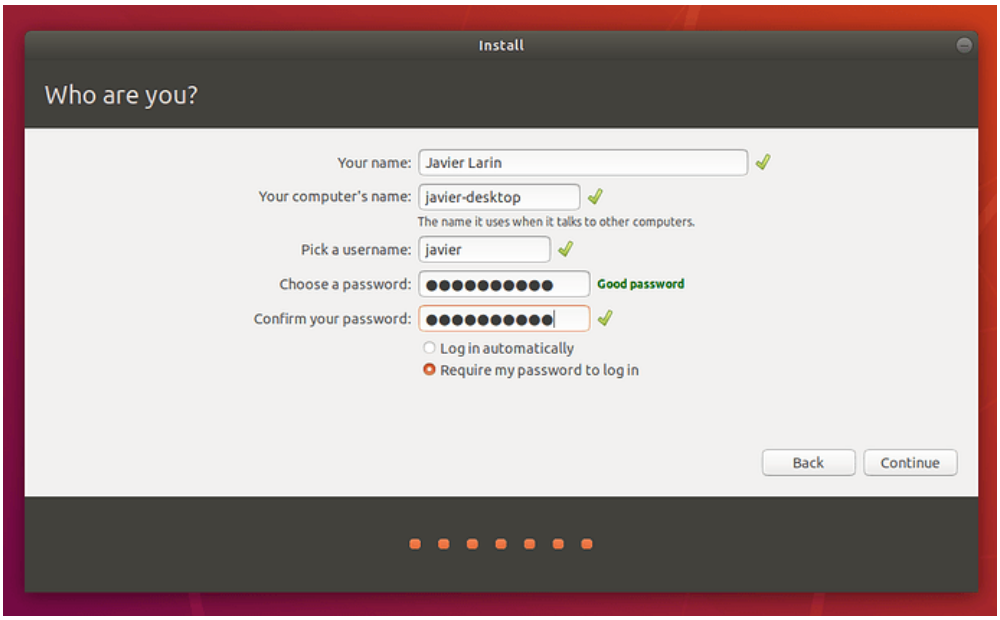


Fig. 2.4 Ubuntu Installation: Login Details

After everything has been installed and configured, a small window (Figure 2.5) will appear asking you to restart your machine. Click on Restart Now and remove either the DVD or USB flash drive when prompted. If you initiated the installation while testing the desktop, you also get the option to continue testing.

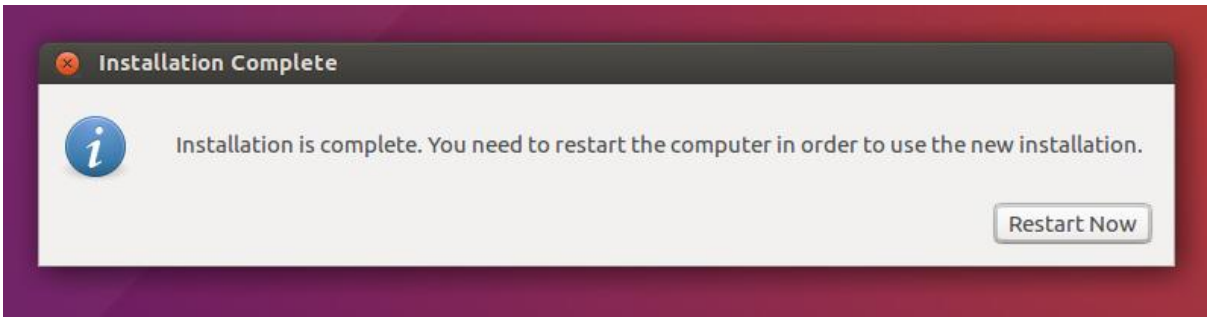


Fig. 2.4 Ubuntu Installation: Restart Window

2.5 Installing Windows

Create USB installation media: Visit Microsoft's Windows 10 download page and select “Download tool now” under the “create Windows 10 installation media” section. Transfer the downloaded installer tool to a USB drive.

Run the installer tool: Open the installer tool by clicking on it. Accept Microsoft's terms, and then select "Create installation media for another PC" on the "What do you want to do?" page. After selecting which language you want, and which edition you want as well (32-bit or 62-bit), you'll be asked what type of media you want to use.

Installing from a USB drive is definitely the preferred option but you can also install from a CD or ISO file. Once you choose your device, the installer tool will download the required files and put them onto your drive.

Use your installation media: Insert your installation media into your device and then access the computer's **BIOS or UEFI**. These are the systems that allow you to control your computer's core hardware. The process of accessing these systems is unique to each device, but the manufacturer's website should be able to give you a helping hand here. Generally, you'll need to press the F2, F12 or Delete keys as your computer boots up.

Change your computer's boot order: Once you have access to your computer's BIOS/UEFI you'll need to locate the settings for boot order. You need the Windows 10 installation tool to be higher up on the list than the device's current boot drive: this is the SSD or HDD that your existing OS is stored on. You should move the drive with the installer files to the very top of the boot order menu**.** Now, when you restart your device the Windows 10 installer should load up first.

Restart your device: Save your settings in the BIOS/UEFI and reboot your device.

Complete the installation: Your device should now load up the Windows 10 installation tool on restart. This will guide you through the rest of the installation process.

Wait for your computer to restart. Once your computer finishes restarting, you'll see a window here with your geographical data. You're now ready to begin setting up your Windows 10 installation. Follow the following steps to complete the installation of Windows 10.

Click Install Now. It's in the middle of the window (Figure 2.5).

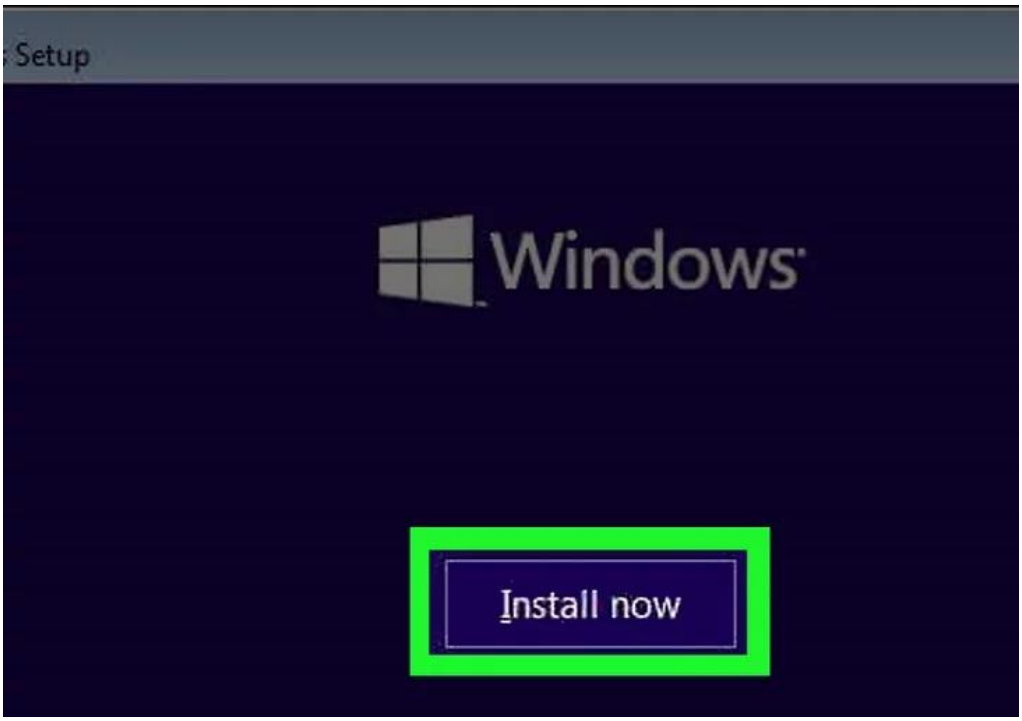


Fig. 2.5 Windows Installation: Install Now

In the next window (Figure 2.6), enter your Windows 10 key, then click Next. If you don't have a Windows 10 key, instead click Skip in the bottom-right corner of the screen.

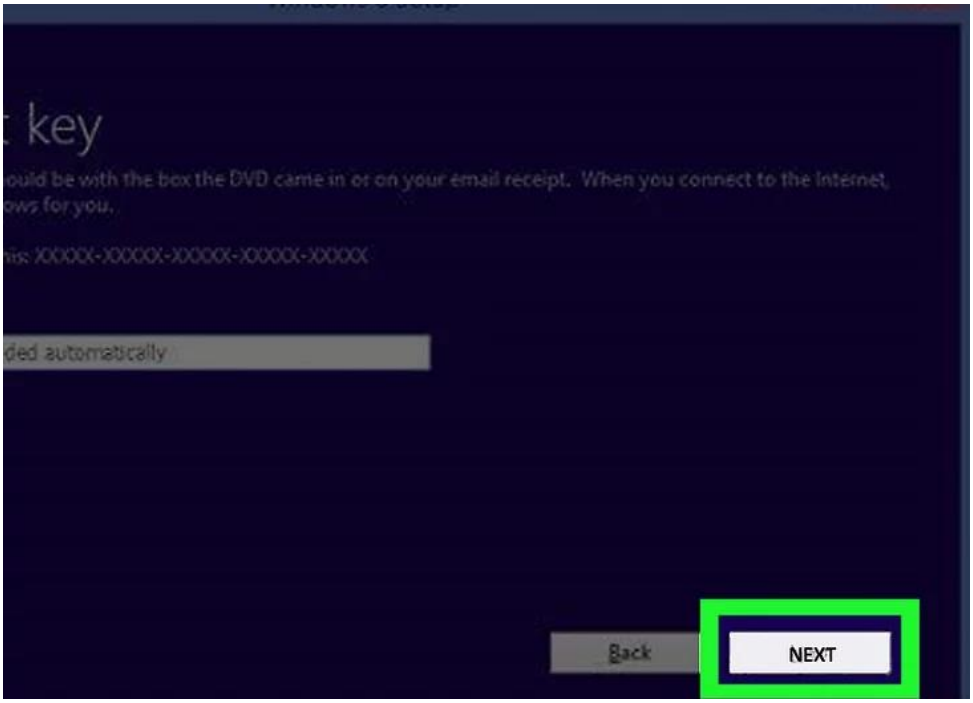


Fig. 2.6 Windows Installation: Enter Key

In the next window (Figure 2.7), click the "Accept" box and click Next. This will indicate that you accept the terms of use.

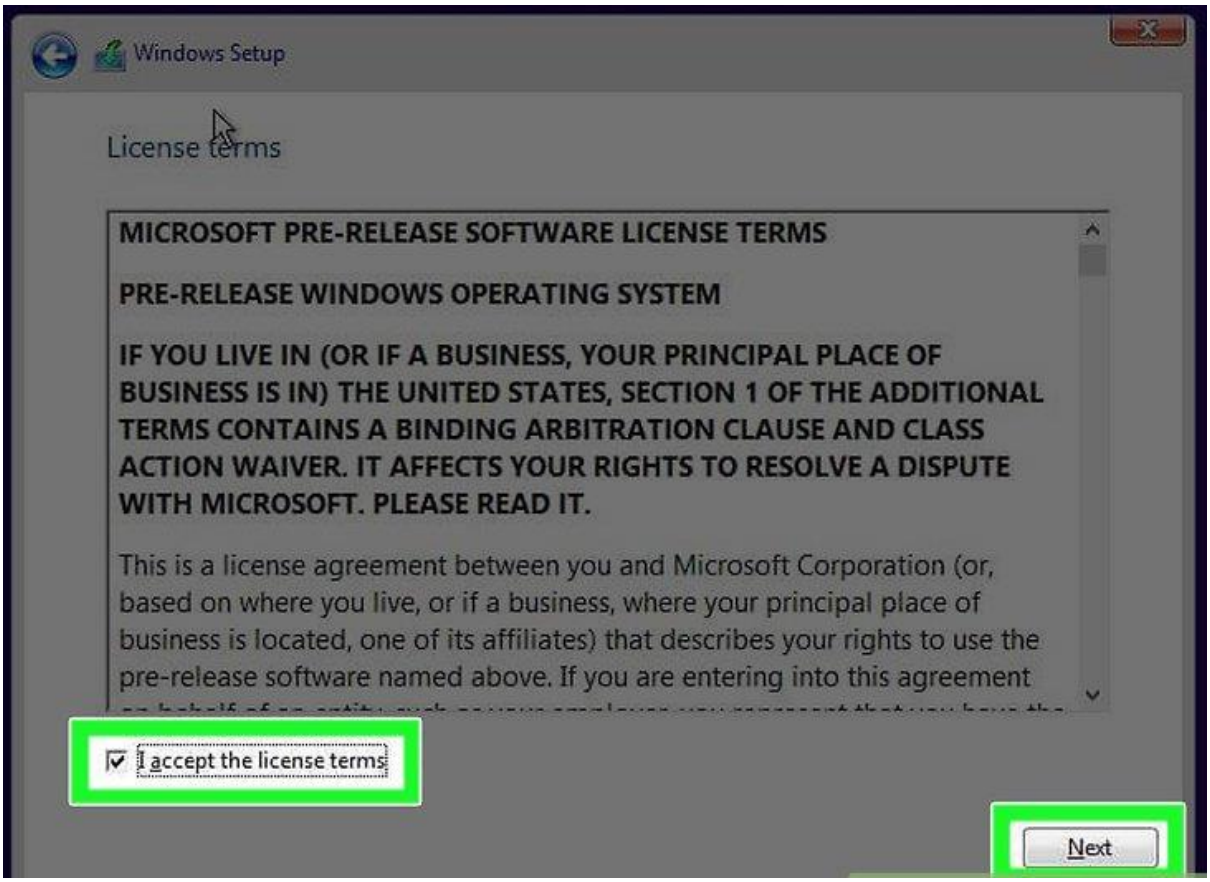


Fig. 2.7 Windows Installation: Accept Terms

In the next window, it will ask: "Which type of installation do you want?". Upgrade option installs Windows 10 while preserving your files, apps, and settings.Wait for Windows 10 to install. This process may take anywhere from half an hour to several hours, depending on your computer's previous operating system and processing speed.

Installing Drivers

Driver updates for Windows 10, along with many devices, such as network adapters, monitors, printers, and video cards, are automatically downloaded and installed through Windows Update. You probably already have the most recent drivers, but if you'd like to manually update or reinstall a driver, follow the steps given below:

Update the device driver

1. In the search box on the taskbar, enter device manager, then select **Device Manager**.
2. Select a category to see names of devices, then right-click (or press and hold) the one you'd like to update. See Figure 2.8.
3. Select **Search automatically for updated driver software**.
4. Select **Update Driver**.
5. If Windows doesn't find a new driver, you can try looking for one on the device manufacturer's website and follow their instructions.

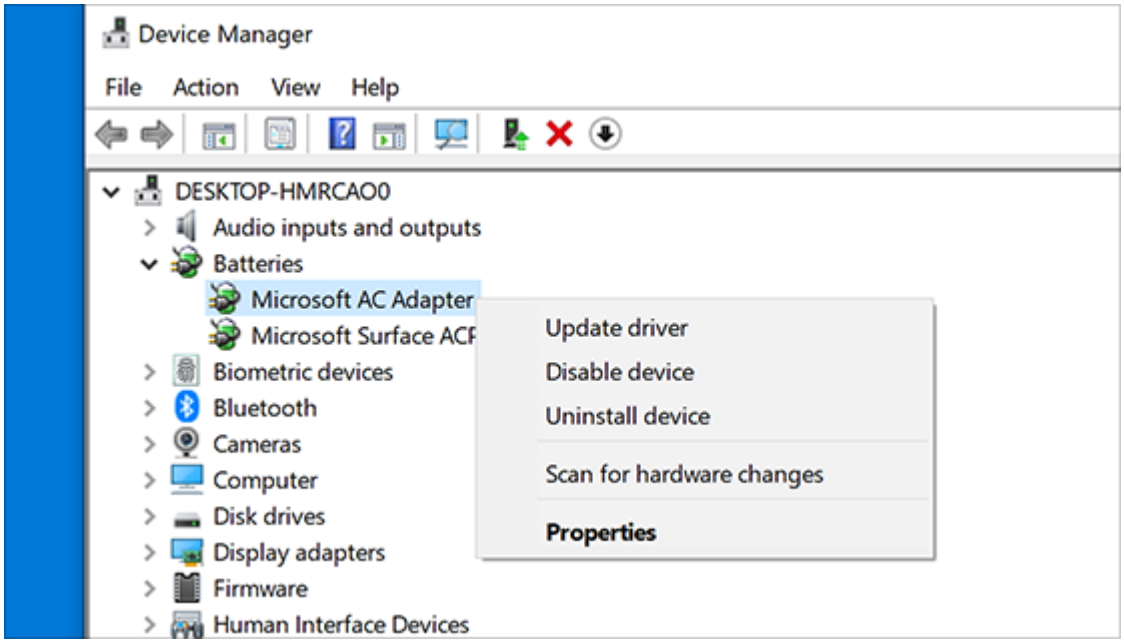


Fig. 2.8 Install Drivers in Windows

2.6 Install New OS in VMware Workstation

Virtual Machine

A Virtual Machine (VM) is a software resource that uses software instead of a physical computer to run programs and deploy apps. One or more virtual "guest" machines run on a physical "host" machine. Each virtual machine runs its own operating system and

functions separately from the other VMs, even when they are all running on the same host.

Follow the following steps to install operating systems on a virtual machine (VMware: commonly used Virtual Machine).

Step 1: Install VMware workstation. Go to **File** > **New Virtual Machine** (see Figure 2.9).

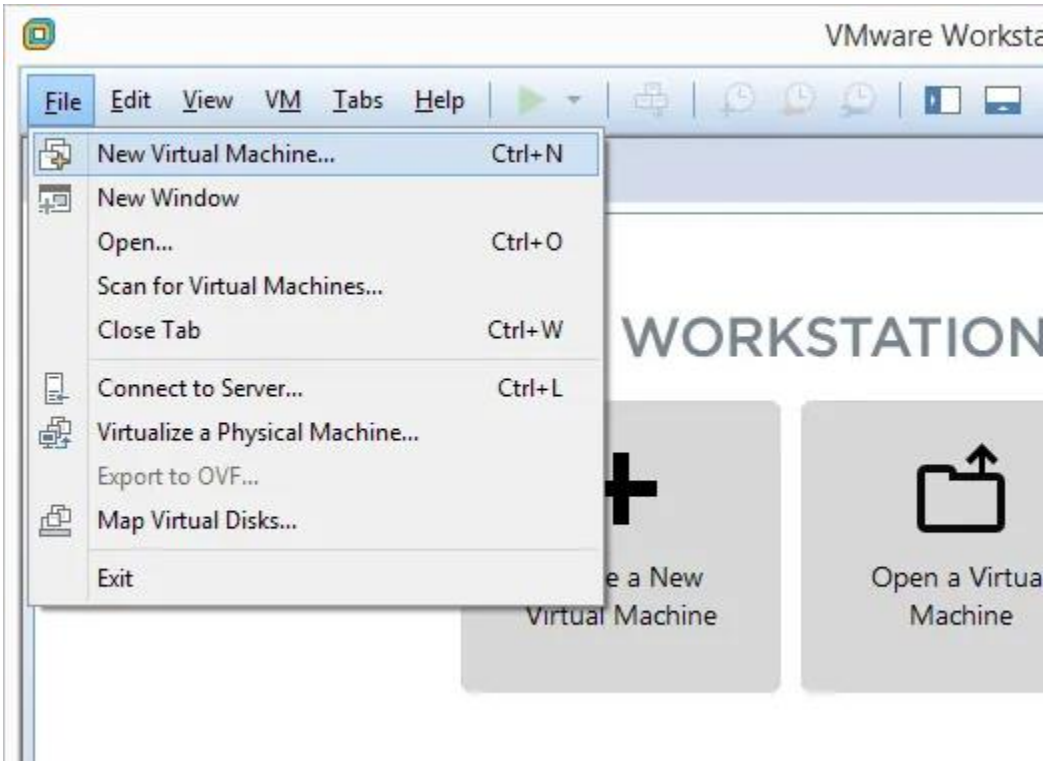


Fig. 2.9 VMware: Select New Virtual Machine

Step 2: Select *Typical (recommended)***,** and then press *Next*(see Figure 2.10).

Fig. 2.10 VMware: Select Recommended Settings

Step 3: Pick how to install the operating system. Select **Installer disc** if the operating system is in the disc drive. Otherwise, choose **Installer disc image file (iso)** if you have a file that contains the OS, like a Windows 10 ISO or an ISO for macOS. Refer to Figure 2.11.

Step 4: Choose precisely where the OS files are located.

Fig. 2.11 VMware: Select Image File

Step 5: Select storage for this virtual machine, and then press **Next**. Refer to Figure 2.12.

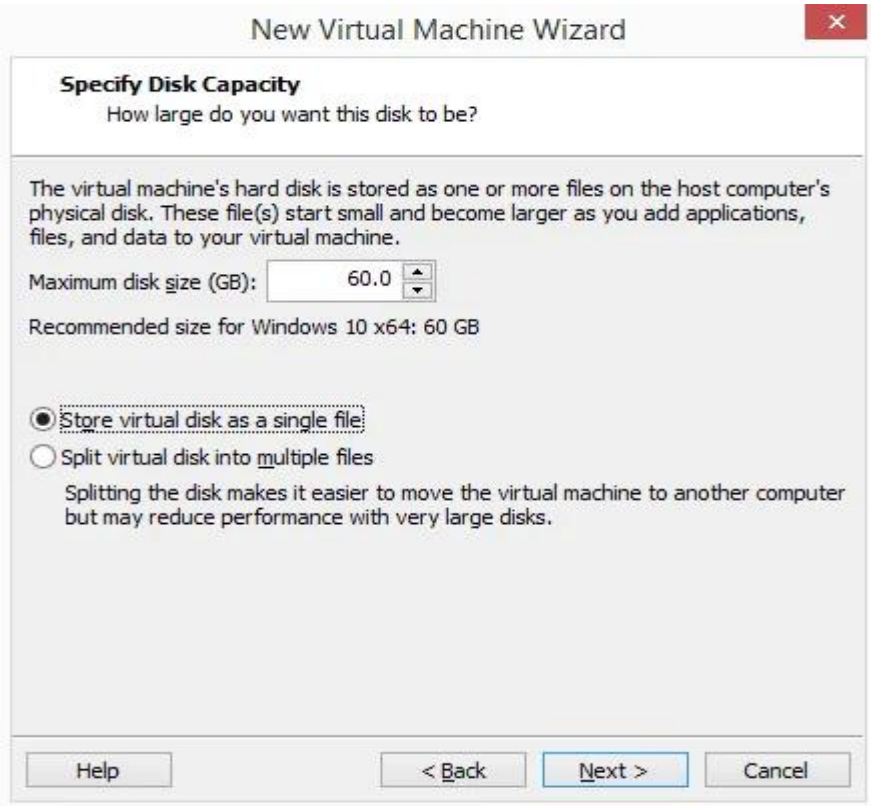


Fig. 2.11 VMware: Select Storage for Virtual Machine

Step 6: Select **Customize Hardware** and make any necessary changes. You can change details about the memory, processors, disc drive, network adapter, USB controller, sound card, printer, and display.

Network Adapter is an important component here that you might end up having to adjust later if your virtual machine doesn't have internet access. There are multiple options, and some may or may not work properly depending on how your host computer is set up and how the guest OS (the VM) works. You can skip this for now to have the default option chosen for you.

Step 7: Choose **Close** to exit the *Hardware* screen, and then press **Finish**. VMware Workstation will create the virtual disk you specified in Step 5 and then turn on the virtual machine automatically. This process might take a while, but you can watch the progress bar for an estimation of when it will finish.

Step 8: Follow the directions for your specific operating system install to add the OS to VMware Workstation.

2.7 Operating Systems Usage

User Interface and commands are used to operate any operating system. We enlist some common ways of using Windows and Linux:

2.7.1 Linux Usage:

1. Cp command

The cp command stands for **copy** is used to **copy files and directories** in Linux System. The syntax for cp command is: cp <Source><Destination>. We can simply use the cp command along with the **source** and **destination** as shown in Figure 2.12.

Fig. 2.12 Use of cp Command

2. Mv command

The mv command stands for **move** is used to **move files and directories** in Linux System. The syntax for mv command is: mv <source><destination>. We can simply use the mv command along with the **source** and **destination** as shown in Figure 2.13.

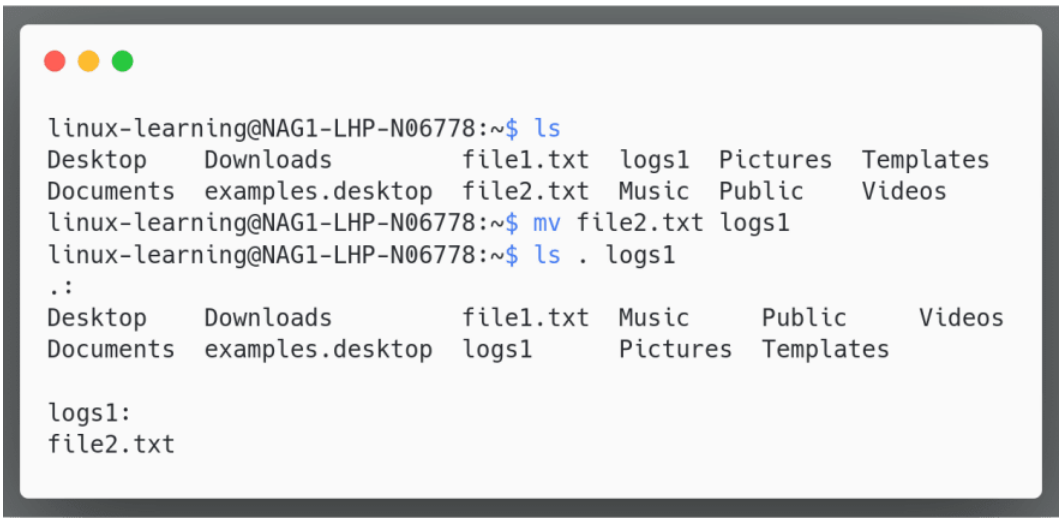


Fig. 2.13 Use of mv Command

3. Mv command for Renaming

The mv command is used for both moving as well as renaming a file/directory. The syntax for mv command for renaming is: mv <current_name><new_name>. To rename a file we just used mv command along with current_name and new_name as shown in Figure 2.14.

Fig. 2.13 Renaming using mv Command

4. Search Files and Folders

find command expressions look like this:

```
find command options starting/path expression
```

The options attribute controls the behaviour and optimization method of the *find* process. The *starting/path* attribute defines the top-level directory where the *find* command in Linux begins the filtering process. The *expression* attribute controls the assessments that scour the directory tree to create output.

Let's break down a Linux find command where we don't just want Linux find file by name:

```
find -O3 -L /var/www/ -name "*.html"
```

It enables the top-level optimization (-O3) and permits *find* to follow symbolic links (-L). The *find* command in Linux searches through the whole directory hierarchy under /var/www/ for files that have .html on the end.

Basic Examples

1. ***find*** **. -name thisfile.txt**

If you need to know how to find a file in Linux called this*file.txt*, it will look for it in current and sub-directories.

2. ***find*** **/home -name *.jpg**

Look for all .jpg files in the /home and directories below it.

3. ***find*** **. -type f -empty**

Look for an empty file inside the current directory.

4. ***find*** **/home -user randomperson-mtime 6 -iname ".db"**

Look for all .db files (ignoring text case) that have been changed in the preceding 6 days by a user called *randomperson*.

2.7.2 Windows Usage

1. Personalization

Windows 10 makes it easy to customize the look and feel of your desktop. To access the **Personalization settings**, right-click anywhere on the desktop, then select **Personalize** from the drop-down menu as shown in Figure 2.14. The Personalization settings will appear.

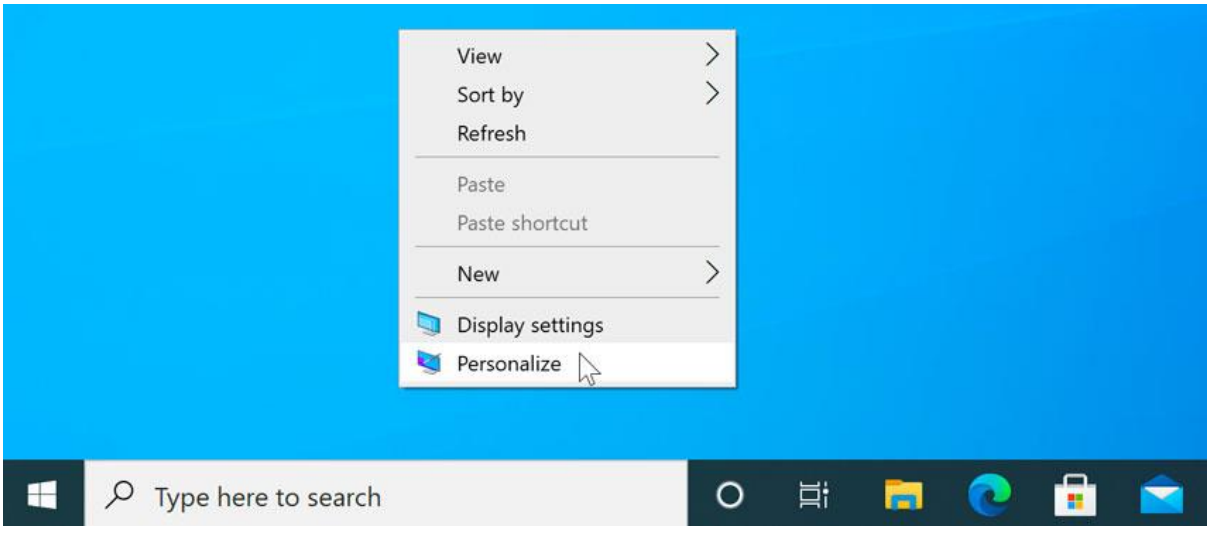


Fig. 2.14 Personalize Settings

2. To change the font size

If you have difficulty seeing the text on your computer, you can increase the **font size**. Increasing the font size will also increase the size of icons and other items on your desktop.

1. Open the **Settings** app, then select **System**.
2. The **Display** options will appear as shown in Figure 2.15. Use the drop-down arrow to scale up or down the size. Note that a larger size may interfere with the way some items appear on the screen.

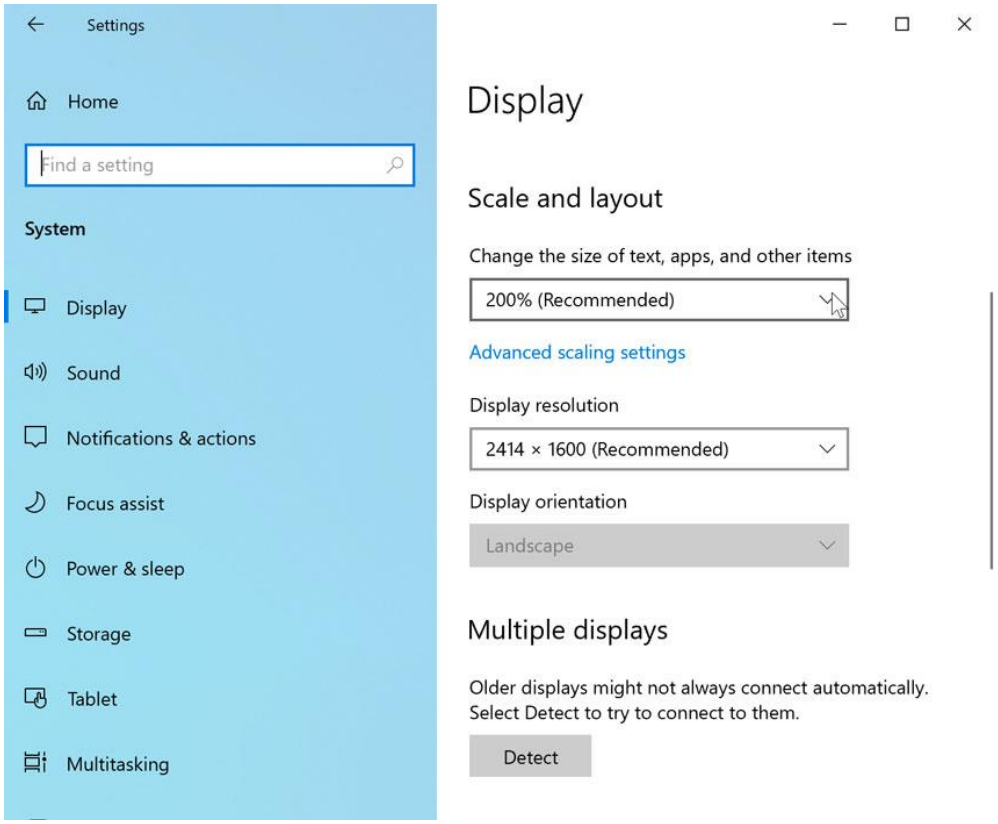


Fig. 2.15 Display Settings

3. Once you've made your selection, the changes will take effect.
4. When you're done, click **Finish**. The ClearType settings will be applied.

3. Windows File Extensions

Windows file names have two parts; the file's name, then a period followed by the extension (suffix). The extension is a three- or four-letter abbreviation that signifies the file type. For example, in **letter.docx** the filename is **letter** and the extension is **docx**. Extensions are important because they tell your computer what icon to use for the file, and what application can open the file. For example, the **doc** extension tells your computer that the file is a Microsoft Word file.

4. Unhide Files and Folders

1. Click on the start menu and type **"Show hidden files and folders,"** then press the enter key or select that option from the list of results as shown in Figure 2.16.

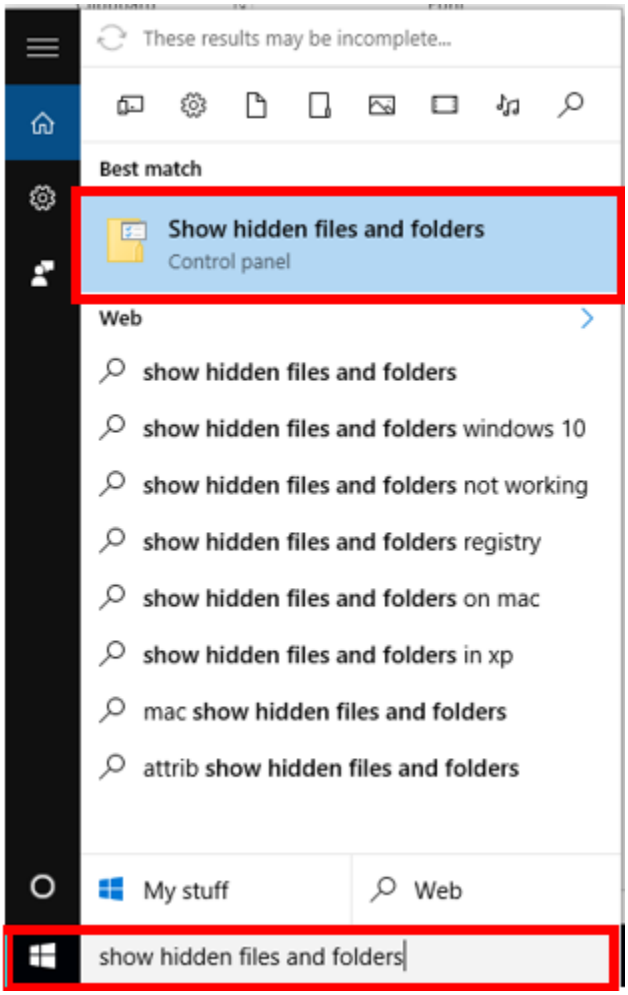


Fig. 2.16 Windows: Show Hidden Files

2. Under **Advanced settings**, select **Show hidden files, folders, and drives**, and then select **OK**.

5. Hide a File or Folder on Windows

To hide a file or folder on Windows, open a Windows Explorer or File Explorer window and locate the file or folder you want to hide. Right-click it and select Properties. Enable the Hidden checkbox on the General pane of the Properties window as shown in Figure 2.17. Click OK or Apply and your file or folder will be hidden.

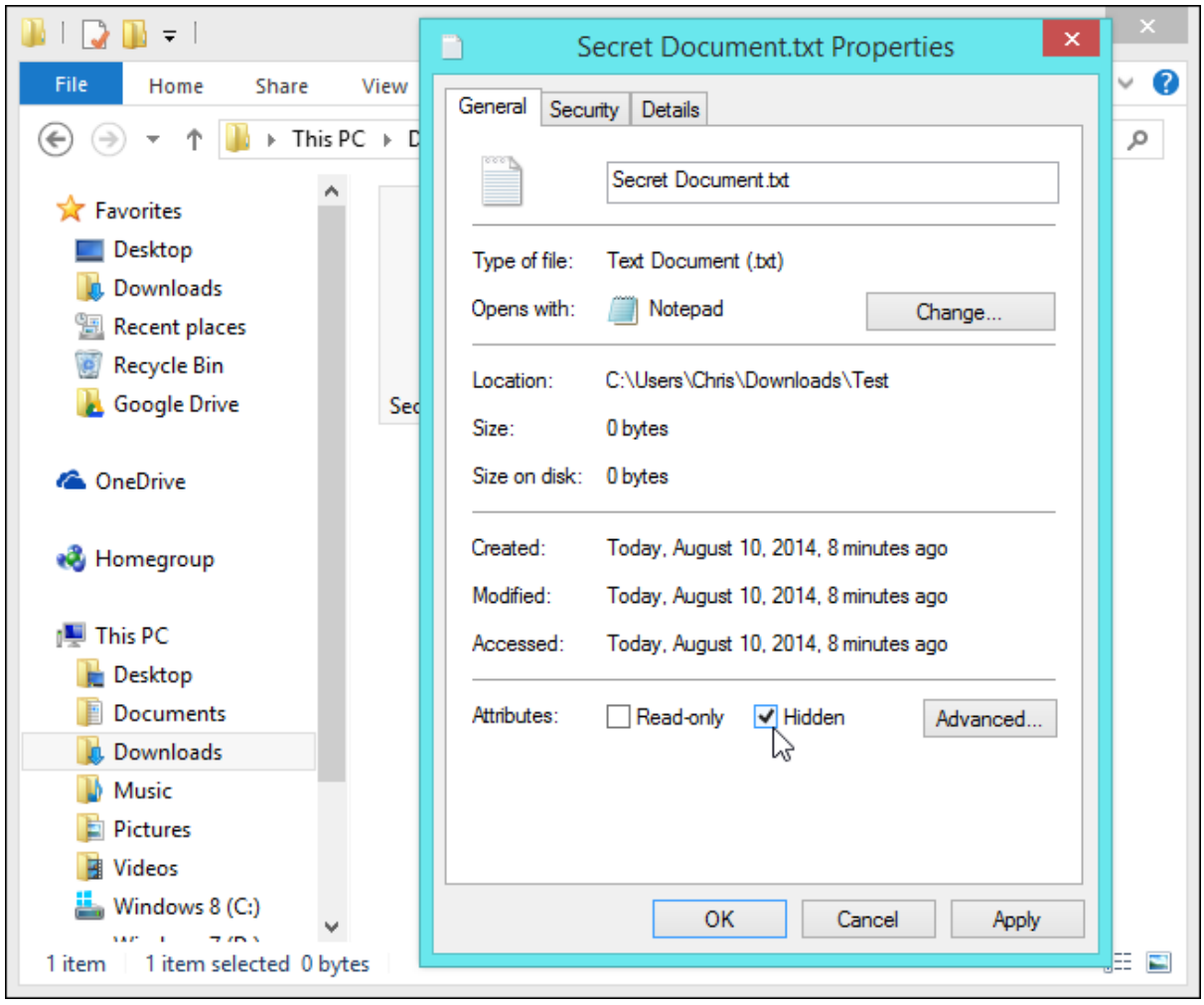


Fig. 2.17 Windows: Hide Files

6. Device Manager

Device Manager is defined as a Control Panel that is an extension of the Microsoft Management Console. It provides users an organized, central view of the Windows-recognized hardware attached to a computer. This includes devices like keyboards, hard disk drives, USB devices etc. It sorts out all hardware, listing them on the basis of various criteria and when a device malfunctions or stops working altogether, it immediately notifies the user. Thus, it can be used to control devices, enable or disable them, configure them and identify conflicts between devices etc.

7.Task Manager

The Windows Task Manager is a powerful tool packed with useful information, from your system's overall resource usage to detailed statistics about each process.

Launching the Task Manager

Windows offers many ways to launch the Task Manager. Press Ctrl+Shift+Esc to open the Task Manager with a keyboard shortcut or right-click the Windows taskbar and select "Task Manager." You can also press Ctrl+Alt+Delete and then click "Task Manager" on the screen that appears or find the Task Manager shortcut in the Start menu.

The first time you launch the Task Manager, you'll see a small, simple window. This window lists the visible applications running on your desktop, excluding background applications. You can select an application here and click "End Task" to close it. This is useful if an application isn't responding—in other words, if it's frozen—and you can't close it the usual way. Figure 2.18 shows a view of a task manager.

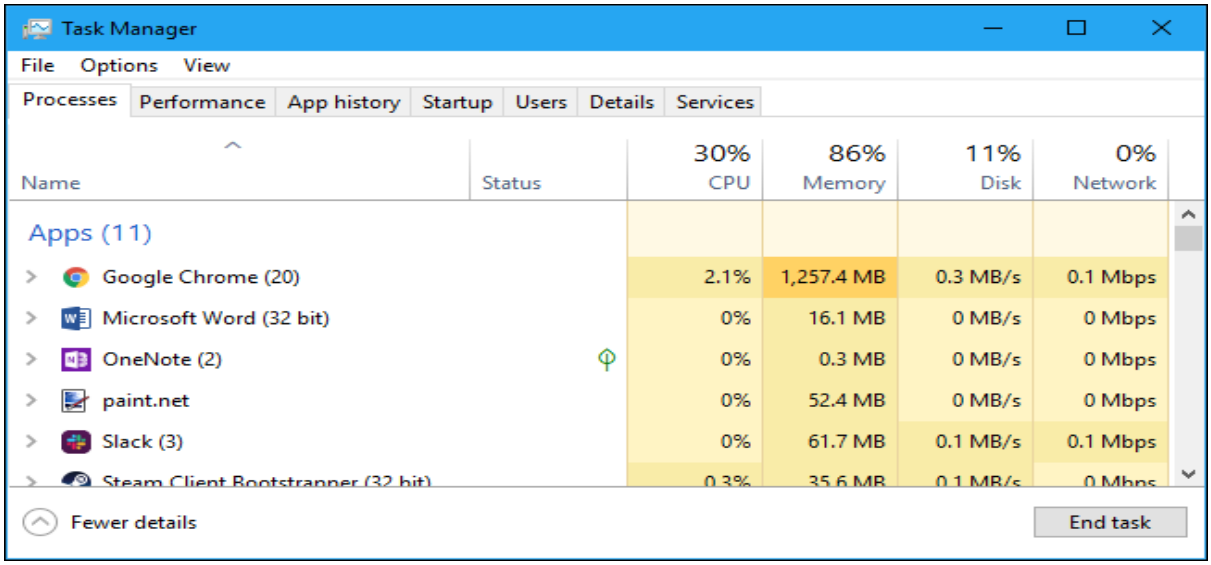


Fig. 2.18 Windows' Task Manager

8. Using the Windows' Command Prompt

Command Prompt is a command line interface for the Windows operating systems. Follow the following steps to open command prompt for Windows:

1. Click Start.
2. In the *Search* or *Run* line, type **cmd** (short for command), and press Enter.

Command prompt in Windows can be used to list contents of a certain directory, move into a directory, run application programs etc. A view of command prompt is given in Figure 2.19.

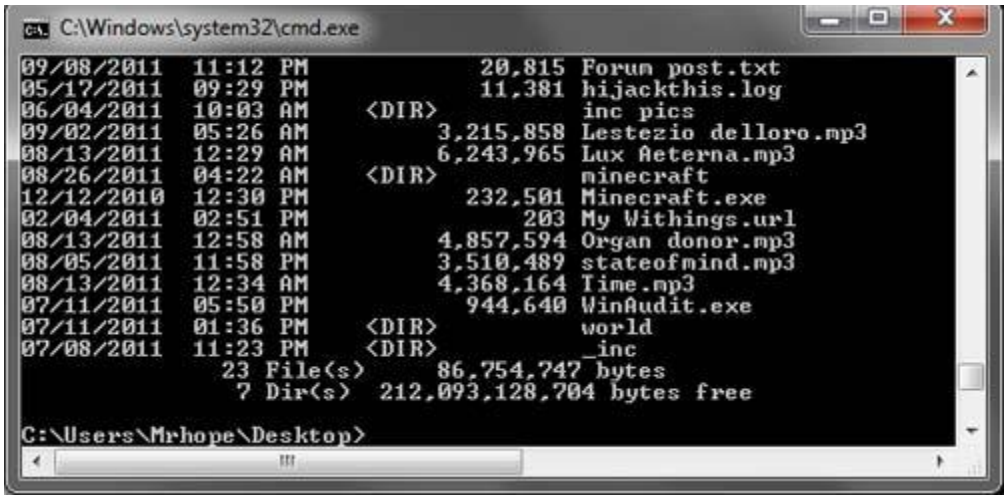


Fig. 2.19 Windows' Command Prompt

Installing an Application Software on Windows

To install programs from a CD or DVD:

1. Insert the program disc into your computer's disc drive or tray, label side up (or, if your computer has a vertical disc slot instead, insert the disc with the label side facing left). The AutoPlay dialog box appears as shown in Figure 2.20.

Fig. 2.20 Windows' AutoPlay Dialog Box

2. Click the option to run Install or Setup. User Account Control may ask if you really want to run this program. If nothing happens when you insert the disc, choose Start→Computer to open the computer window; double-click the icon for your DVD or CD drive; and then double-click a program named Setup or Install.
3. If the installer offers a language selection, choose yours; likewise, agree to the end-user license agreement, if requested. Many installers require you to accept an end-user license agreement (EULA).
4. If the installer offers Express or Custom installation options, choose the Express option to let the installer set up the program without further input from you. The

Custom option or Advanced Settings allows you to specify where to install the program and, perhaps, which parts of the program to install. Some installers provide other options to install documentation and other programs or to register the program.

5. As the installer program continues to run and display dialog boxes, click the Next or Continue button. You proceed to the next dialog box each time you click.
6. Click the Finish or Close button in the last step of the installer program to complete the process.

In a few cases, the setup program may ask you to restart Windows 7. In this case, you don't have to restart immediately, but you won't be able to use the new program until you do restart. The new program may appear automatically at the bottom of the first screen of the Start menu. Look under All Programs as well. Newly installed programs are highlighted in colour. Some installers add a program icon to the desktop.

Many programs try to connect to the Internet for updates during installation or when you run the installed program. The first time you run a program, you may be asked if you want to register the program or configure some aspect of the program. Go with default (assumed) responses, if you're not sure.

2.8 Update vs Upgrade

Updating involves making changes to an app or an operating system is such a way that it doesn't affect its core structure. So, most of the frequent changes made to your computer like bug fixes, security patches, adding support for drivers and newer hardware, etc. can be termed as an Update. An update is often small in size, and it might take a couple of minutes to perform one. The word 'small' is relative. For instance, an update meant for a single app can range from a few kilobytes to a couple of megabytes in size, while an update for an operating system can go up to a few hundred megabytes.

When a set of changes made to a software are significant and substantial enough, it is called an Upgrade. That's why making a switch from Ubuntu 16.04 to Ubuntu 17.04

would be called an upgrade, not update. An upgrade mostly includes important changes to the GUI and a variety of new features and options which are not in the existing version of a software or operating system. And as you might have guessed, its size can go up to several gigabytes.

Updating Windows

To manage update options and see available updates, select Check for Windows updates. Or select the Start button, and then go to Settings > Update & Security > Windows Update.

Upgrade from Windows 7 to Windows 10

Users can either do an in-place upgrade on same computer or a new clean install on a new PC. In both methods simply use the Windows 7 Serial key when asked and it will work perfectly. Make sure you have enough free disk space before upgrading. Check the minimum free disk space required for Windows 10 upgrade.

In-place Upgrade

In place upgrade helps to upgrade your existing Windows7 or Windows8 system to the latest windows10 by keeping all your programs and data migrated to the latest version. However, we recommend you to take the backup of your data before starting the procedure.

Step 1: Download & run media creation tool.

Step 2: Select the option '**upgrade this PC**'. When you run the media creation tool it will give you two options as shown in Figure 2.21. Select the first one "Upgrade this PC now".

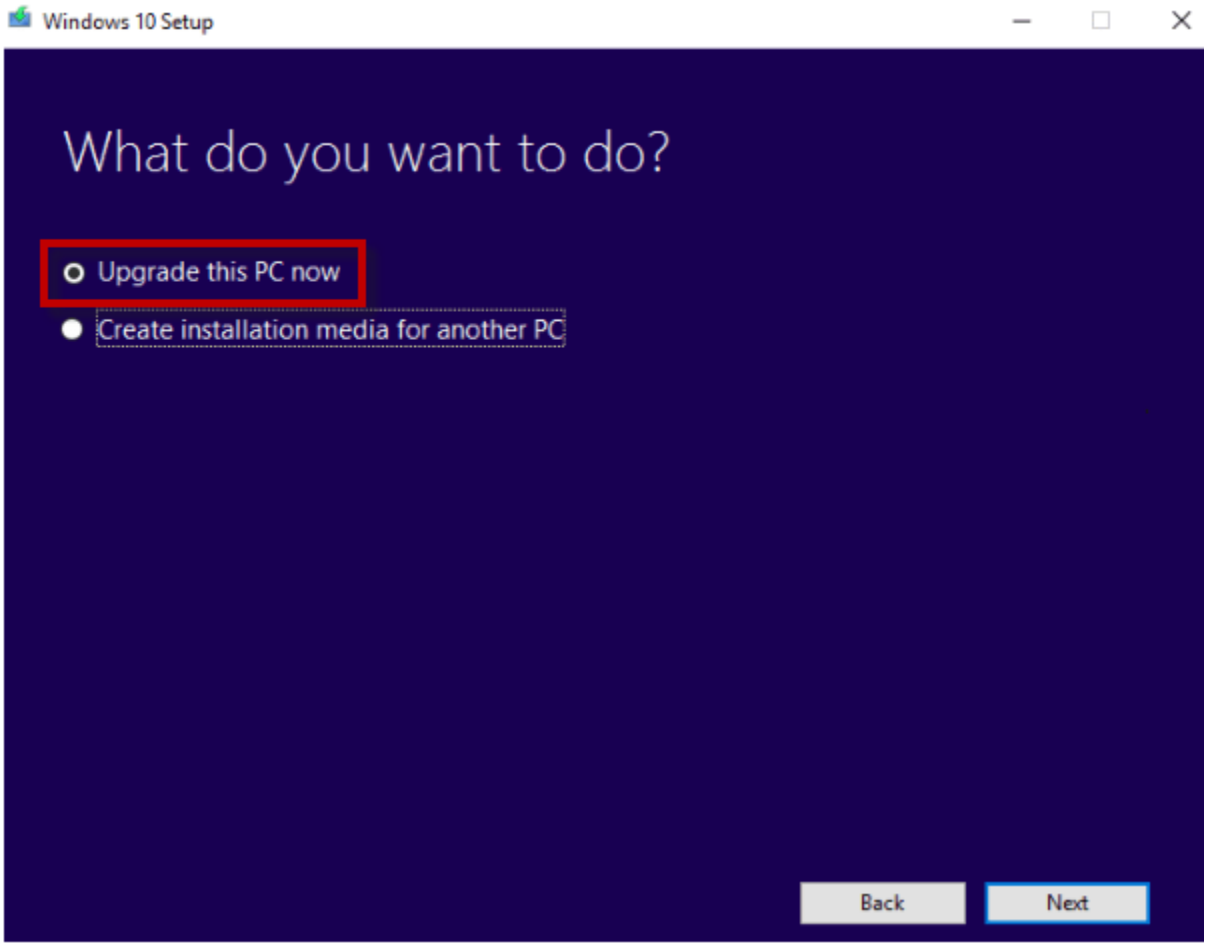


Fig. 2.21 Windows' Media Creation Tool: Upgrade Options

Step 3: Wait while it downloads the required files and then click 'Install' button.

It may take some time in downloading Windows 10 files from the internet. So, it depends on your internet speed that how much time will it take. If you are using a pirated or cracked version then it may ask you to enter the product key. Otherwise, it will not ask for any product / license key and upgrade process will go smoothly. After completion of download, click on the '**Install**' button.

Step 4: Select 'Keep the personal files and apps' option as shown in Figure 2.22. After clicking on the **install** button, it will ask for what to keep. You can choose data only, data & apps or nothing. By default, it will select the '**data & apps**' option.

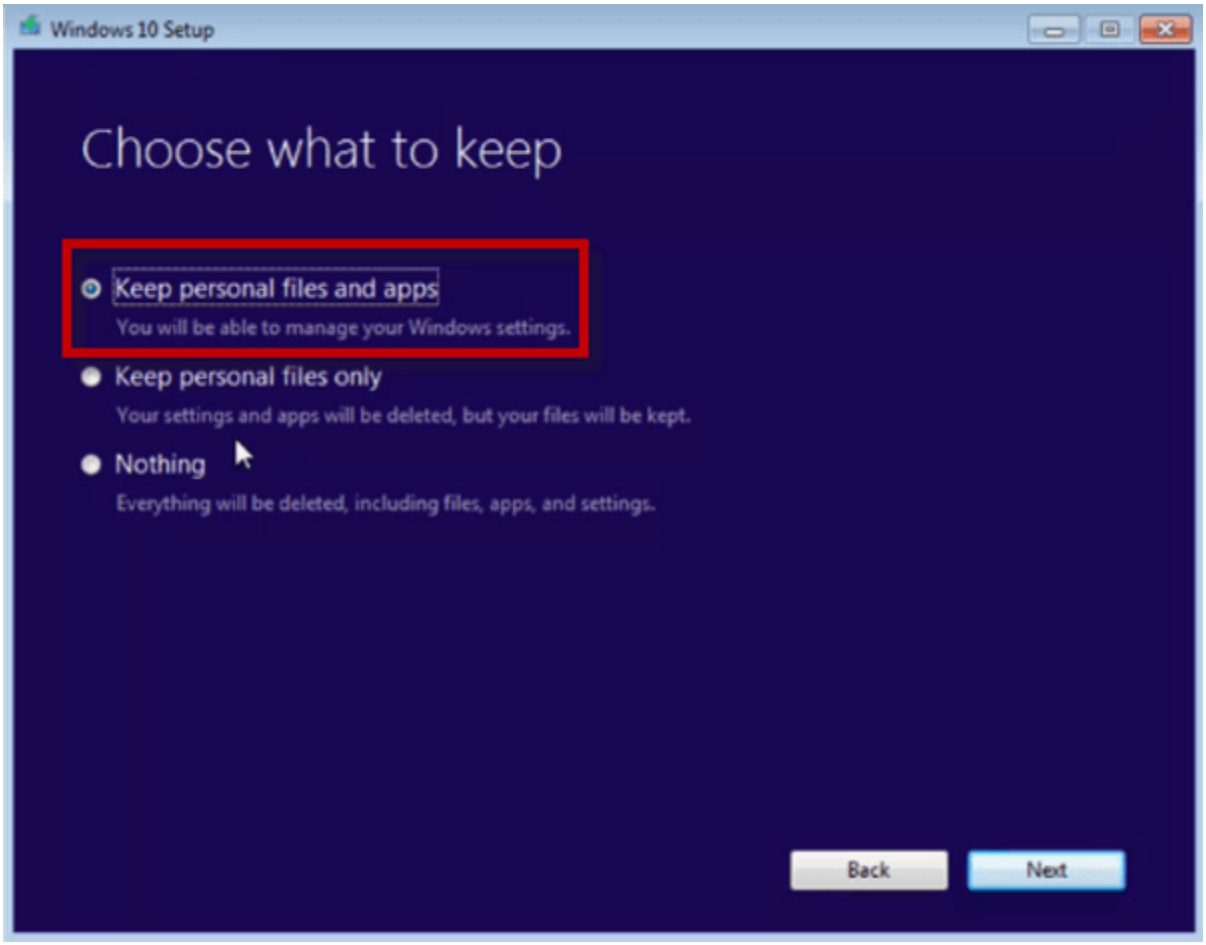


Fig. 2.21 Windows' Media Creation Tool: Backup Options

Step 5: Wait while upgrade process finishes. Click on the **next** button and it will start the upgrade process and may restart the system several times.

Step 6: If everything goes smooth you are welcome by Welcome screen of Windows 10. It will ask you to set the privacy settings first. Set the privacy settings and you are ready to go.

Key Points

- A software is a set of instructions or programs instructing a computer to do specific tasks.
- System software is an interface between application software and system.
- Application software is the type of software which runs as per user request. It runs on a platform which is provided by the system software.
- An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.
- A firmware is a tangible electronic component with embedded software instructions, such as a BIOS.
- A Virtual Machine (VM) is a compute resource that uses software instead of a physical computer to run programs and deploy apps.
- The cp command stands for copy is used to copy files and directories in Linux System.
- The mv command stands for **move** is used to **move files and directories** in Linux System.
- It provides users an organized, central view of the Windows-recognized hardware attached to a computer.
- Updating involves making changes to an app or an operating system is such a way that it doesn't affect its core structure.
- When a set of changes made to a software are significant and substantial enough, it is called an Upgrade.

Exercise

Select the most appropriate option.

1. Windows is a

   a. Operating Systems b. Firmware c. Application Software d. Both a and b

2. VMware is a

a. Operating Systems b. Firmware c. Application Software d. Both a and b

3. Which of the following OS is open source?

   a. Linux b. Windows c. MSDOS d. None of these

4. cp command is used to

   a. move files b. copy files c. delete files d. none of these

5. mv command is used to

   a. move files b. copy files c. delete files d. none of these

Answer the following questions.

1. Define Software.
2. Differentiate between applications software and system software?
3. Describe functionalities of an operating system.
4. Define Firmware.
5. Define Virtual Machine.
6. Define Task Manager.
7. Describe command prompt and its usage.
8. What is the purpose of Device Manager in Windows?
9. Differentiate Update and Upgrade.

Practical Tasks

1. Install Windows 10 on the given PC.
2. Replace Windows Installation with the Ubuntu Installation.
3. Install Ubuntu on VMware Workstation.
4. Use Command Prompt to copy the contents of a file into another file.
5. Uninstall an application program from Windows using Control Panel.
6. Upgrade Windows 7 to Windows 10 and make sure to keep all programs and data in the Windows 10.

Chapter 3
Computer Networks

After Studying this chapter, you will be able to

- understand the basic types of computer network (LAN, WAN, MAN, PAN etc.).
- describe the basic knowledge of components used in networks.
- know and understand the communication media: unshielded twisted-pair (UTP), shielded twisted pair (STP), fiber optics and coaxial cable. wireless media.
- describe terminologies used in computer network (data rate, Baud, Bit, Kbps, attenuation etc.).
- understand types of cables used in networking.
- differentiate straight and cross cable and understanding of connectors: RJ45, RJ-11, BNC.
- make cross wired and straight through connection.
- know about use of cable tester to test cable connector.
- understand fixing a cable and connector issue.
- know and understand of network components: modems, hubs, switches, bridges, routers etc.
- describe computer network topologies and differentiate active and passive network components.
- configure router settings and know about connecting cables with networking devices.
- know about network protocols (TCP / IP), OSI model, IPv4, IPv6, physical addresses.
- know about Static and dynamic IP and understanding of DHCP.
- know Setting of: IP Address (IP4/IP6) & subnet mask, gateway, classes of IP addressing.
- know about commands used in testing network (ping etc).
- check the Internet connectivity using router interface.

3.1 Computer Network

A computer network is a group of computers linked to each other that enables them to communicate with each other and share their resources, data, and applications.

3.1.1 Types of Computer Networks

A computer network can be categorized by their size. A **computer network** is mainly of **four types** (as shown in Figure 3.1):

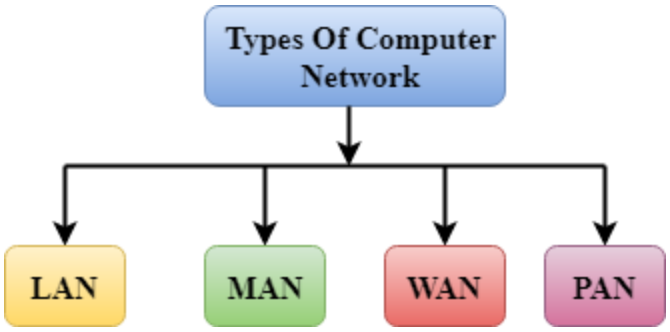


Fig. 3.1 Types of Computer Networks

- LAN (Local Area Network)
- PAN (Personal Area Network)
- MAN (Metropolitan Area Network)
- WAN (Wide Area Network)

LAN (Local Area Network)

Local Area Network is a group of computers connected to each other (as shown in Figure 3.2) in a small area such as building, office.

- LAN is used for connecting two or more personal computers through a communication medium such as twisted pair, coaxial cable, etc.
- It is less costly as it is built with inexpensive hardware such as hubs, network adapters, and ethernet cables.
- The data is transferred at an extremely faster rate in Local Area Network.
- Local Area Network provides higher security.

Fig. 3.2 Local Area Network

PAN (Personal Area Network)

Personal Area Network is a network arranged within an individual person, typically within a range of 10 meters.

- Personal Area Network is used for connecting the computer devices of personal use is known as Personal Area Network.
- Personal Area Network covers an area of **30 feet**.
- Personal computer devices that are used to develop the personal area network are the laptop, mobile phones, media player and play stations as shown in Figure 3.3.

Fig. 3.3 Personal Area Network

MAN (Metropolitan Area Network)

A metropolitan area network is a network that covers a larger geographic area by interconnecting a different LAN to form a larger network (see Figure 3.4).

- Government agencies use MAN to connect to the citizens and private industries.
- In MAN, various LANs are connected to each other through a telephone exchange line.
- The most widely used protocols in MAN are RS-232, Frame Relay, ATM, ISDN, OC-3, ADSL, etc.
- It has a higher range than Local Area Network (LAN).

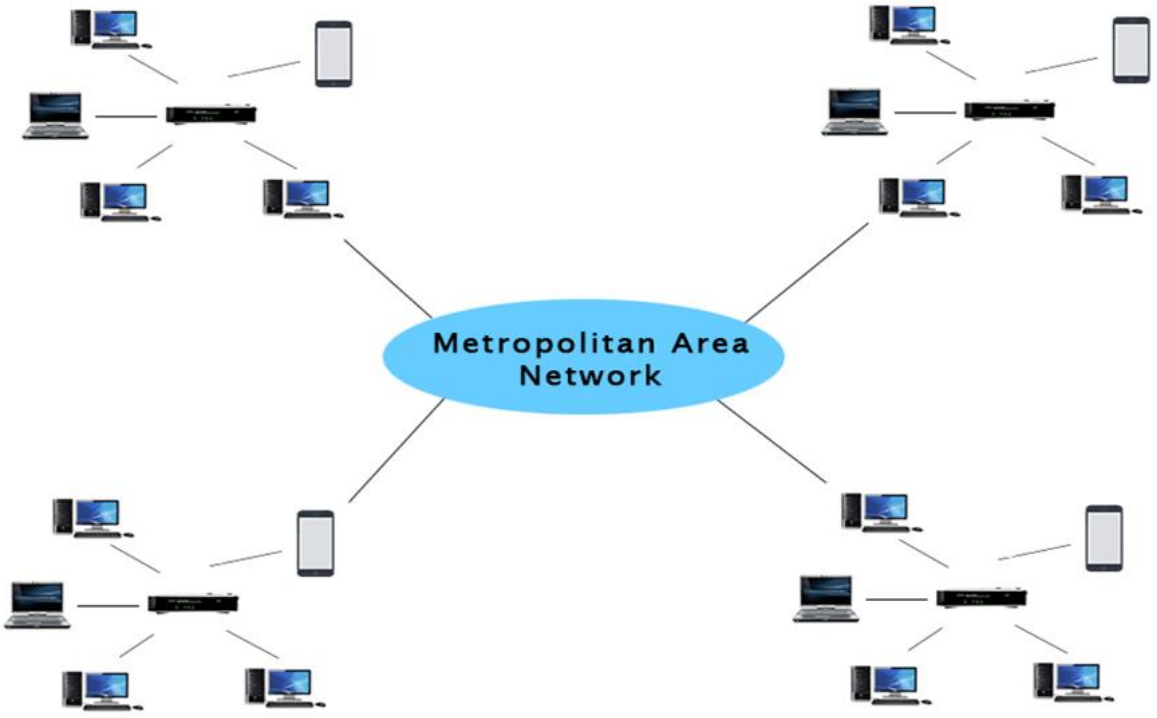


Fig. 3.4 Metropolitan Area Network

WAN (Wide Area Network)

A Wide Area Network (Figure 3.5) is a network that extends over a large geographical area such as states or countries.

- A Wide Area Network is quite bigger network than the LAN.

- A Wide Area Network is not limited to a single location, but it spans over a large geographical area through a telephone line, fibre optic cable or satellite links.
- The internet is one of the biggest WAN in the world.
- A Wide Area Network is widely used in the field of business, government, and education.

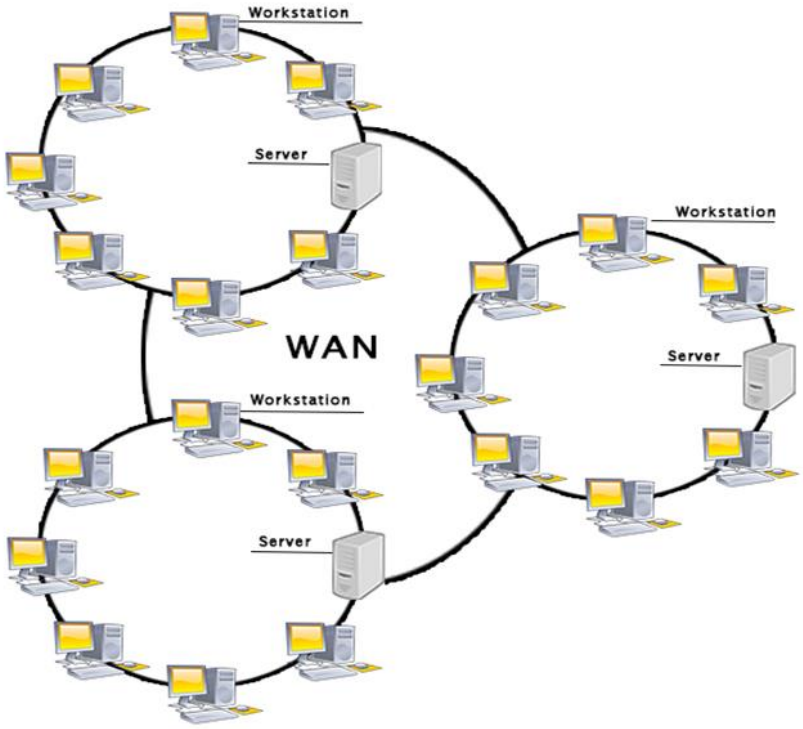


Fig. 3.5 Wide Area Network

3.1.2 Importance of Computer Networking

In the world of Information Technology (IT), information is a building block for effective communication. Communication is medium that helps us to drive our day to day professional and personal operations. Computer networks helps us to achieve the following:

- Provides best way of business communication.
- Streamline communication.
- Cost-effective resource sharing.
- Improving storage efficiency and volume.
- Cut costs on software.

- Cut costs on hardware.
- Utilizes Centralized Database.
- Increase in efficiency.
- Optimize convenience and flexibility.
- Allows file sharing.
- Sharing of peripherals and internet access.
- Network gaming.
- Voice over IP (VoIP).
- Media Center Server.
- Centralize network administration, meaning less IT support.
- Flexibility.
- Allowing information sharing.
- Supporting distributed processing.
- User communication.
- Overcoming geographic separation.

In general, we can see "**Communication" and "Sharing",** that is all networking is about in simple terms. It enables everything that we do today. In the current scenario, the business would not work without computer networking and our personal and social life will also feel like it has been stopped.

3.1.3 Basic Computer Network Components

Computer networks share common devices, functions, and features including servers, clients, transmission media, shared data, shared printers and other hardware and software resources, network interface card (NIC), local operating system (LOS), and the network operating system (NOS).

Servers - Servers are computers that hold shared files, programs, and the network operating system. Servers provide access to network resources to all the users of the network. There are many different kinds of servers, and one server can provide several functions. For example, there are file servers, print servers, mail servers, communication

servers, database servers, fax servers and web servers, to name a few. Sometimes it is also called host computer, servers are powerful computer that store data or application and connect to resources that are shared by the user of a network.

Clients - Clients are computers that access and use the network and shared network resources. Client computers are basically the customers (users) of the network, as they request and receive services from the servers. These days, it is typical for a client to be a personal computer that the users also use for their own non-network applications.

Transmission Media - Transmission media are the facilities used to interconnect computers in a network, such as twisted-pair wire, coaxial cable, and optical fiber cable. Transmission media are sometimes called transmission medium channels, links or lines.

Shared data - Shared data are data that file servers provide to clients such as data files, printer access programs and e-mail.

Shared printers and other peripherals - Shared printers and peripherals are hardware resources provided to the users of the network by servers. Resources provided include data files, printers, software, or any other items used by clients on the network.

Network Interface Card - Each computer in a network has a special expansion card called a network interface card (NIC). The NIC prepares (formats) and sends data, receives data, and controls data flow between the computer and the network. On the transmit side, the NIC passes frames of data on to the physical layer, which transmits the data to the physical link. On the receiver's side, the NIC processes bits received from the physical layer and processes the message based on its contents.

Local Operating System - A local operating system allows personal computers to access files, print to a local printer, and use one or more disk and CD drives that are located on the computer. Examples are MS-DOS, Unix, Linux, Windows 2000, Windows 98, Windows XP etc. The network operating system is the software of the network. It serves a similar purpose that the OS serves in a stand-alone computer

Network Operating System - The network operating system is a program that runs on computers and servers that allows the computers to communicate over the network.

Hub - Hub is a device that splits a network connection into multiple computers. It is like a distribution center. When a computer requests information from a network or a specific computer, it sends the request to the hub through a cable. The hub will receive the request and transmit it to the entire network. Each computer in the network should then figure out whether the broadcast data is for them or not.

Switch - Switch is a telecommunication device grouped as one of computer network components. Switch is like a Hub but built in with advanced features. It uses physical device addresses in each incoming messages so that it can deliver the message to the right destination or port.

Unlike a hub, switch doesn't broadcast the received message to entire network, rather before sending it checks to which system or port should the message be sent. In other words, switch connects the source and destination directly which increases the speed of the network.

Router - The device used to connect a LAN with an internet connection is called Router. When you have two distinct networks (LANs) or want to share a single internet connection to multiple computers, we use a Router.

LAN Cable - A local area network cable is also known as data cable or Ethernet cable which is a wired cable used to connect a device to the internet or to other devices like computer, printers, etc.

3.2 Network Cable Types and Specifications

To connect two or more computers or networking devices in a network, network cables are used. There are three types of network cables; coaxial, twisted-pair, and fiber-optic.

Coaxial cable

This cable contains a conductor, insulator, braiding, and sheath. The sheath covers the braiding, the braiding covers the insulation, and the insulation covers the conductor. The following image (Figure 3.6) shows these components.

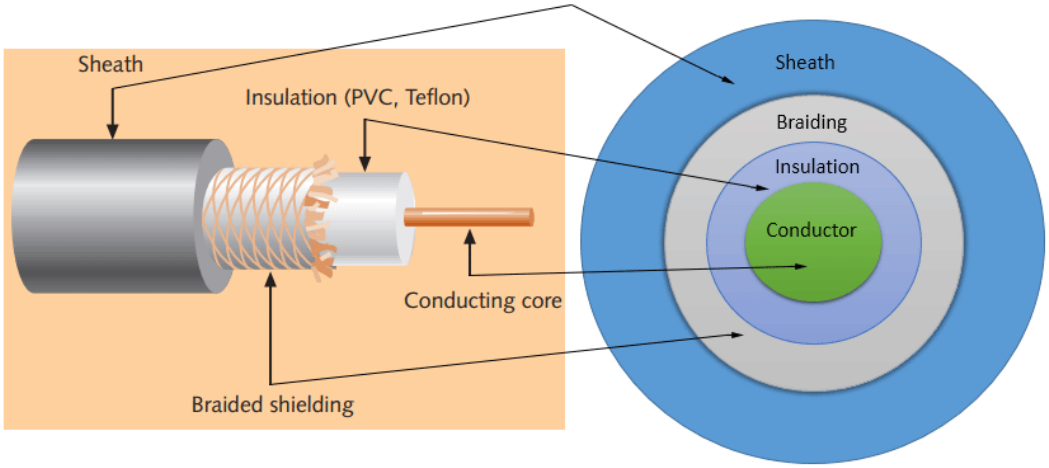


Fig. 3.6 Coaxial Cable

Twisted-pair Cables

The twisted-pair cable was primarily developed for computer networks. This cable is also known as **Ethernet cable**. Almost all modern LAN computer networks use this cable. This cable consists of color-coded pairs of insulated copper wires. Every two wires are twisted around each other to form pair. Usually, there are four pairs. Each pair has one solid color and one stripped color wire. Solid colors are blue, brown, green, and orange. In stripped color, the solid color is mixed with the white color. Based on how pairs are stripped in the plastic sheath, there are two types of twisted-pair cable; UTP and STP.

In the **UTP (*Unshielded twisted-pair*) cable**, all pairs are wrapped in a single plastic sheath.

In the **STP (*Shielded twisted-pair*) cable**, each pair is wrapped with an additional metal shield, then all pairs are wrapped in a single outer plastic sheath.

The following image (Figure 3.7) shows both types of twisted-pair cables.

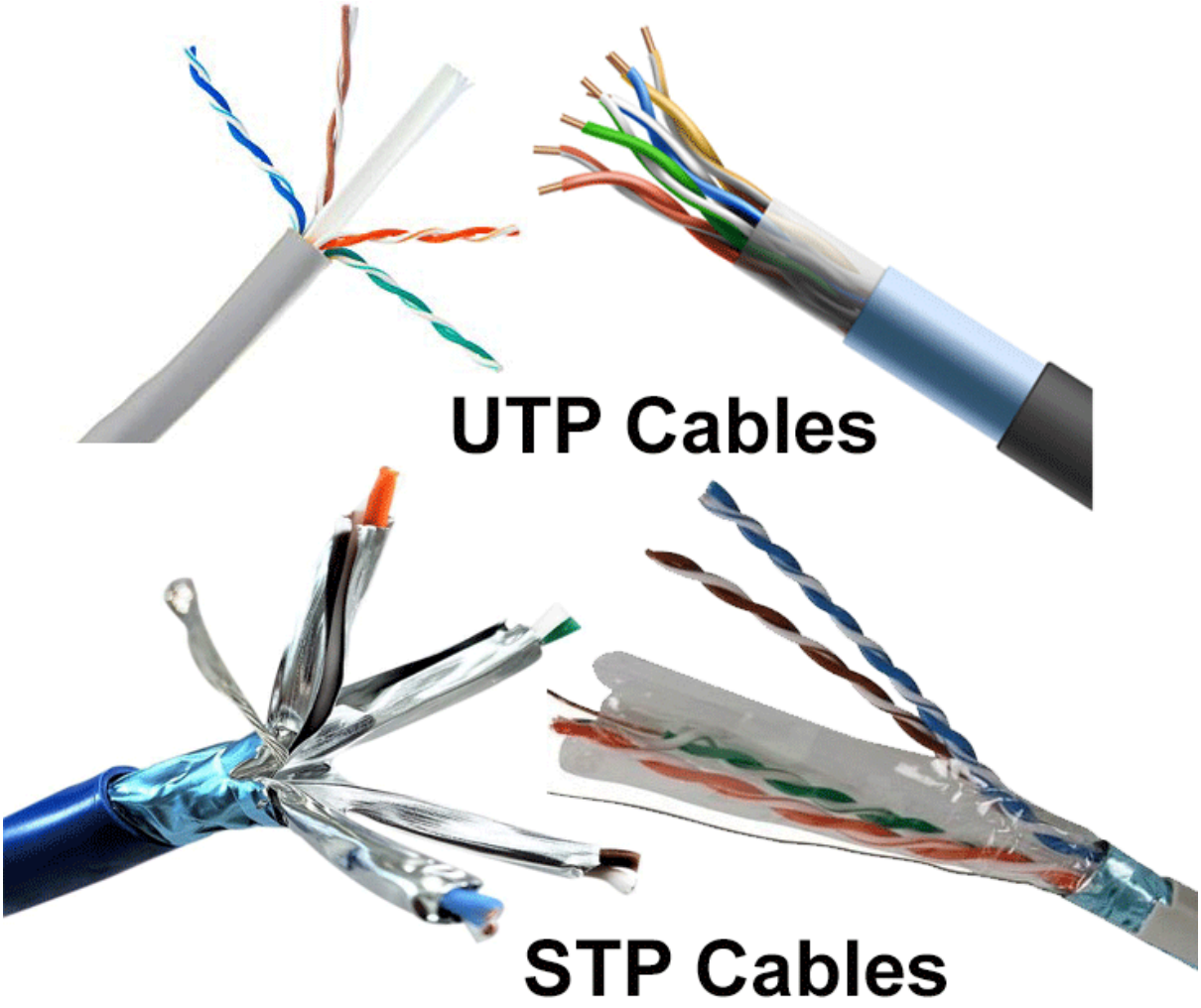


Fig. 3.7 Twisted-pair Cables

Fiber Optic Cable

This cable consists of a core, cladding, buffer, and jacket. The core is made from thin strands of glass or plastic that can carry data over a long distance. The core is wrapped in the cladding; the cladding is wrapped in the buffer, and the buffer is wrapped in the jacket. Fiber optic cable is completely immune to EMI and RFI. This cable can transmit data over a long distance at the highest speed. It can transmit data up to 40 kilometres at the speed of 100Gbps. Fiber optic uses light to send data. It reflects light from one endpoint to another. Based on how many beams of light are transmitted at a given time, there are two types of fiber optical cable; SMF and MMF as shown in Figure 3.8.

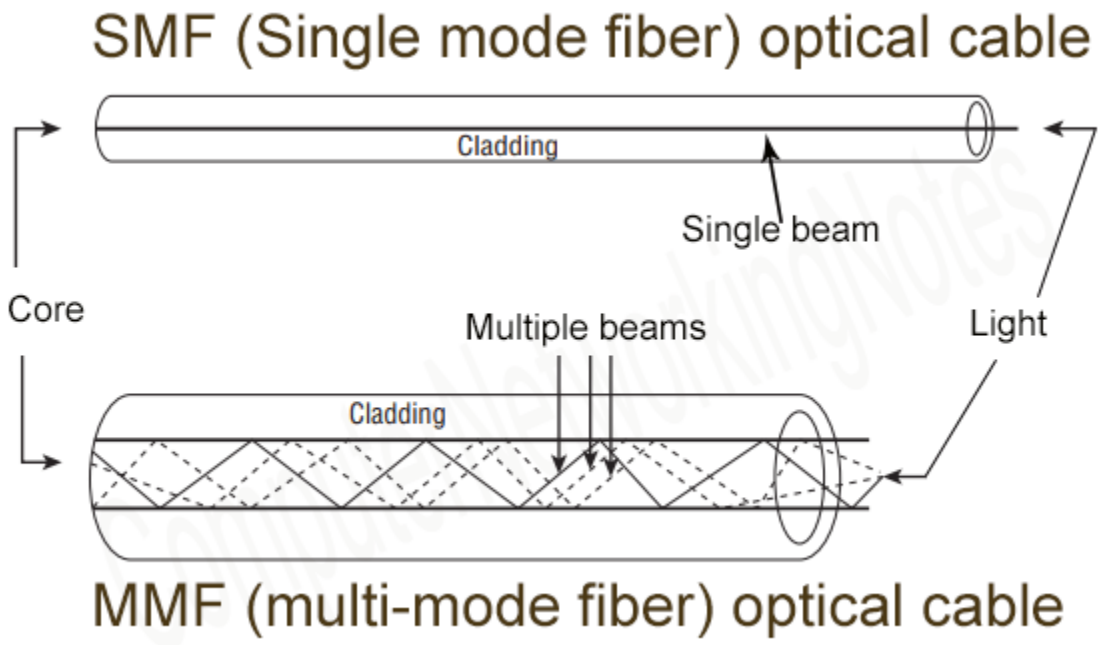


Fig. 3.8 Fiber Optic Cable

What is an Ethernet cable?

An Ethernet cable is a network cable used for high-speed wired network connections between two devices. This network cable is made of a four-pair cable, which consists of twisted pair conductors. It is used for data transmission at both ends of the cable, which is called RJ45 connector.

The Ethernet cables are categorized as CAT5, CAT5e, CAT6, and UTP cable. CAT5 cable can support a 10/100 Mbps Ethernet network while Cat5e and Cat 6cable to support Ethernet network running at 10/100/1000 Mbps.

What is straight through Cable?

Straight-through cable is a type of CAT5 with RJ-45 connectors at each end, and each has the same pin out as shown in Figure 3.9. It is in accordance with either the T568A or T568B standards. It uses the same color code throughout the LAN for consistency. This type of twisted-pair cable is used in LAN to connect a computer or a network hub such as a router. It is one of the most common types of network cable.

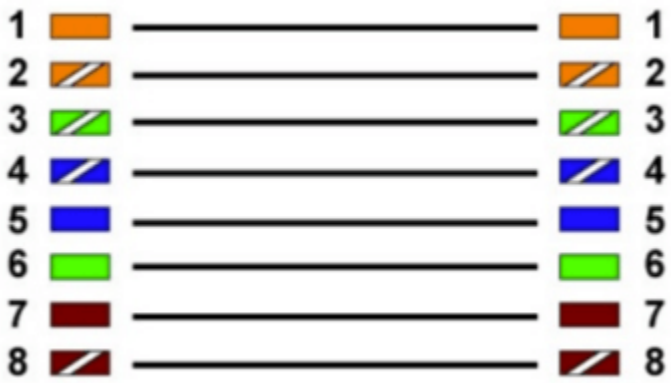


Fig. 3.9 Straight Through Cable Connection

What Is Crossover Cable?

A Crossover cable is a type of CAT5 where one end isT568A configuration and the other end as T568BConfiguration. In this type of cable connection, Pin 1 is crossed with Pin 3, and Pin 2 is crossed with Pin 6 as shown in Figure 3.10. Crossover cable is used to connect two or more computing devices. The internal wiring of crossover cables reverses the transmission and receive signals. It is widely used to connect two devices of the same type: e.g., two computers or two switches to each other.

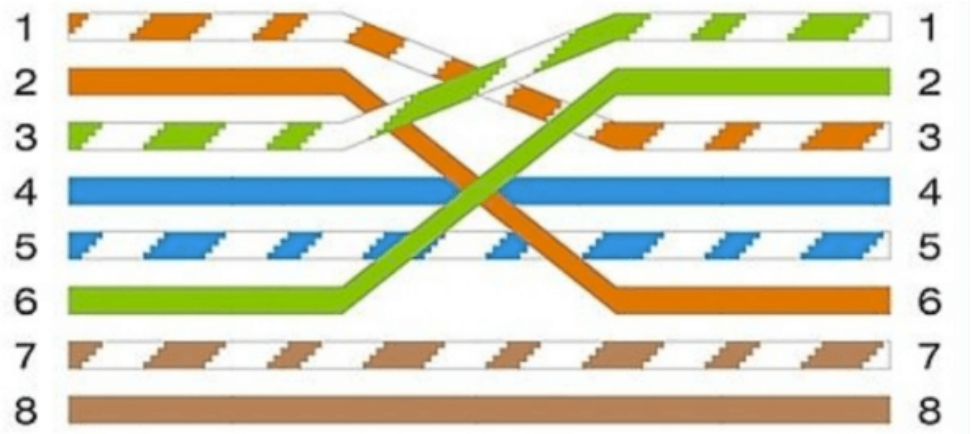


Fig. 3.10 Cross Over Cable Connection

RJ 45 and RJ 11:

RJ is the acronym for Registered Jack and refers to the industry standard character that cable connectors usually start with. RJ45 and RJ11 are the 2 most command connector standards used universally. The main difference between these two is in where they are

actually used. RJ45 jacks are used in networking (Ethernet LAN), where you connect computers or other network elements to each other. RJ11 is the cable connector that is being used in telephone sets.

Using Cable Tester to Check Connection

Figure 3.11 shows a cable tester. It is used to check connection between two ends of a cable. Follow the steps given below to check an Ethernet wire using cable tester.

Fig. 3.11 Cable Tester

1. Plug one end of the cable into the transmit jack. The transmit jack on the tester maybe be labelled "TX".
2. Plug the other end of the cable into the receiver jack. The receiver jack may be labelled "RX" on the device. Some testers have a detachable receiver you can use to test the cable across different rooms.
3. Check the lights on the tester. Most testers will have 2 sets of 8 LED lights that correspond to the 8 pins on the transmitting and receiving end of the Ethernet cable. There will also be a G light for the ground. It will test each pin one at a time. If all 8 pins light up on both ends, the cable is good. If any of the lights do

not light up on either end, this indicates a short in the cable. Don't worry if the G light doesn't light up. If the lights on either end flash out of order, this indicates that you are testing a crossover cable. As long as all 8 lights flash, the cable is good.

Fixing a RJ45 Connector

1. Strip the cable back 1 inch (25 mm) from the end.
2. Untwist and straighten the wires inside of the cable.
3. Arrange the wires into the right order.
4. Cut the wires into an even line ½ inch (13 mm) from sheathing.
5. Insert the wires into the RJ-45 connector.
6. Stick the connector into the crimping part of the tool and squeeze twice.
7. Remove the cable from the tool and check that all of the pins are down.

Wireless vs Wired Network

A wireless network allows devices to stay connected to the network but roam untethered to any wires. Access points amplify Wi-Fi signals, so a device can be far from a router but still be connected to the network. When you connect to a Wi-Fi hotspot at a cafe, a hotel, an airport lounge, or another public place, you're connecting to that business's wireless network.

A wired network uses cables to connect devices, such as laptop or desktop computers, to the Internet or another network. A wired network has some disadvantages when compared to a wireless network. The biggest disadvantage is that your device is tethered to a router. The most common wired networks use cables connected at one end to an Ethernet port on the network router and at the other end to a computer or other device.

Bit Rate:

The speed of the data is expressed in bits per second (bits/s or bps). The data rate R is a function of the duration of the bit. Rate is also called channel capacity C.

Teacher Notes:

- Give a demonstration of speedtest.net to students.

Baud Rate

Baud rate refers to the number of signal or symbol changes that occur per second. A symbol is one of several voltage, frequency, or phase changes. NRZ (Non Return to Zero) binary has two symbols, one for each bit 0 or 1, that represent voltage levels. In this case, the baud or symbol rate is the same as the bit rate. However, it's possible to have more than two symbols per transmission interval, whereby each symbol represents multiple bits. With more than two symbols, data is transmitted using modulation techniques. By using multiple symbols, multiple bits can be transmitted per symbol. For example, if the symbol rate is 4800 baud and each symbol represents two bits, that translates into an overall bit rate of 9600 bits/s. Normally the number of symbols is some power of two. If N is the number of bits per symbol, then the number of required symbols is $S = 2^N$.

Attenuation

Attenuation in computer networking is the loss of communication signal strength that is measured in decibels (dB). As the rate of attenuation increases, the transmission, such as an email a user is trying to send or a phone call, becomes more distorted. Attenuation occurs on computer networks because of:

- Range – over longer distances both wired and wireless transmissions gradually dissipate in strength
- Interference – radio interference or physical obstructions, such as walls, dampen communication signals on wireless networks
- Wire size – thinner wires suffer from more attenuation than thicker wires on wired networks

3.3 Network Devices

1. Repeater – A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network. An important point to be noted about repeaters is that they do not amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2-port device.

2. Hub – A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. In other words, the collision domain of all hosts connected through Hub remains one. Also, they do not have the intelligence to find out the best path for data packets which leads to inefficiencies and wastage.

3. Bridge – A bridge operates at the data link layer. A bridge is a repeater, with add on the functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2-port device.

4. Switch – A switch is a multiport bridge with a buffer and a design that can boost its efficiency (a large number of ports imply less traffic) and performance. A switch is a data link layer device. The switch can perform error checking before forwarding data, which makes it very efficient as it does not forward packets that have errors and forward good packets selectively to the correct port only.

5. Routers – A router is a device like a switch that routes data packets based on their IP addresses. The router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets. Router divide broadcast domains of hosts connected through it.

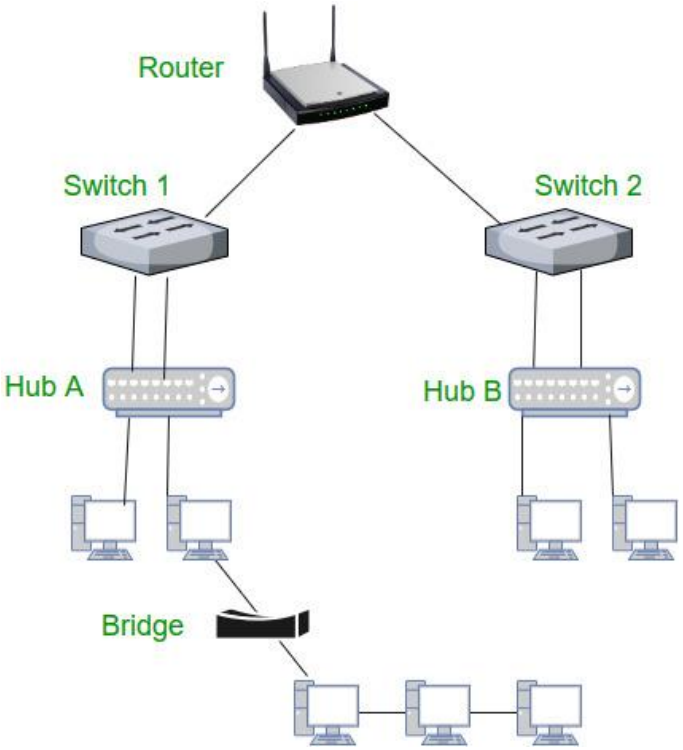


Fig. 3.12 Network Devices

6. Gateway – A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models. They basically work as the messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switches or routers. Gateway is also called a protocol converter.

7. NIC – NIC (Network Interface Card) is a network adapter that is used to connect the computer to the network. It is installed in the computer to establish a LAN. It has a unique id that is written on the chip, and it has a connector to connect the cable to it. The cable acts as an interface between the computer and router or modem. NIC card is a layer 2 device which means that it works on both physical and data link layer of the network model.

3.3.1 Router Interfaces

Routers can have many different types of connectors; from Ethernet, Fast Ethernet, and Token Ring to Serial and ISDN ports. Some of the available configurable items are logical addresses (IP, IPX), media types, bandwidth, and administrative commands. Interfaces are configured in interface mode which is available from global configuration mode after logging in. Generally, three types of interfaces are provided to manage a Network switch:

Simple Network Management Protocol*:*

Also known as SNMP, it is a protocol that facilitates the exchange of management information between network devices. ***SNMP*** queries can determine the health of a network or the status of a particular device.

Command Line Interface:

Command Line Interface (CLI***)*** is an exhaustive set of commands that can be accessed via serial console, Telnet, and Secure Shell.

The Web Interface:

User can monitor and configure the switch from a ***browser***, by entering a URL.

3.3.2 Wireless Devices and their Roles

There are three major "modes" a Wi-Fi device can use. These modes define the role a Wi-Fi device has in the network, and networks must be built out of combinations of devices operating in these different modes. How the devices are configured depends on the types of connections you want to use between parts of the network.

The three wireless roles are

1. Wireless Clients (Station)

Devices such as computers, tablets, and phones are common clients on a network. When you are accessing a wireless hotspot, or the router in your home or office, your device is the client. This client mode is also known as "station mode" as well.

2. Access Points (Master)

Most wireless networks are made using Access Points - devices that host and control the wireless connection for laptops, tablets, or smart phones. If you use Wi-Fi in your home or office, it is most likely through an Access Point. When a router is set up as an AP, it is said to be in “Master” or “Infrastructure” mode.

An AP is sometimes a stand-alone device that bridges between a wireless and wired (Ethernet) network, or is part of a router. APs can cover a range of areas with a wireless signal, depending on the power of the device and the type of antenna. There are also some APs that are weatherproof, designed to be mounted outdoors.

3. Ad-Hoc Node (Mesh)

Some wireless devices (laptops, smart phones, or wireless routers) support a mode called Ad-Hoc. This allows those devices to connect together directly, without an Access Point in-between controlling the connection. This forms a different type of network - in Ad-Hoc mode, all devices are responsible for sending and receiving messages to the other devices - without anything else in between. In an Ad-Hoc network, every device must be in this role, and using the same configuration to participate.

3.4 Network Topologies

A Network Topology is the arrangement with which computer systems or network devices are connected to each other. Topologies may define both physical and logical aspect of the network. Both logical and physical topologies could be same or different in a same network.

Point-to-Point

Point-to-point networks contains exactly two hosts such as computer, switches or routers, servers connected back-to-back using a single piece of cable. Figure 3.13 shows a Point-to-point network. Often, the receiving end of one host is connected to sending end of the other and vice-versa.

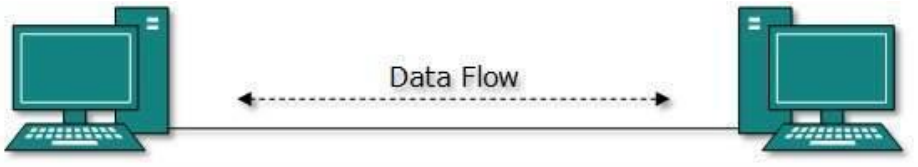


Fig. 3.13 Point-to-point Networks

If the hosts are connected point-to-point logically, then may have multiple intermediate devices. But the end hosts are unaware of underlying network and see each other as if they are connected directly.

Bus Topology

In case of Bus topology, all devices share single communication line or cable. Bus topology may have problem while multiple hosts sending data at the same time. It is one of the simple forms of networking where a failure of a device does not affect the other devices. But failure of the shared communication line can make all other devices stop functioning.

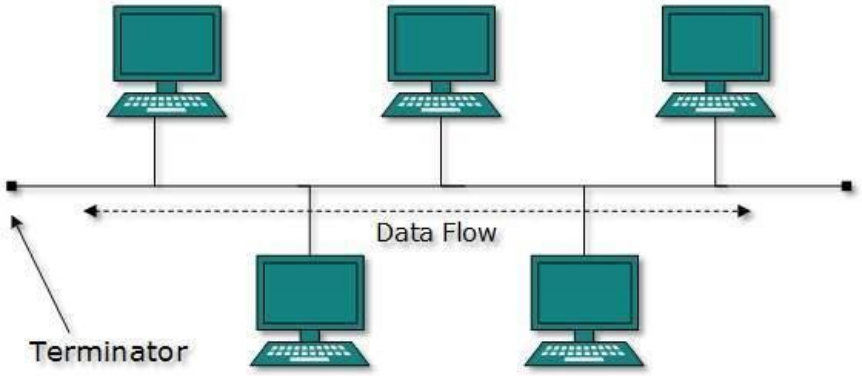


Fig. 3.14 Bus Topology

Both ends of the shared channel have line terminator as shown in Figure 3.14. The data is sent in only one direction and as soon as it reaches the extreme end, the terminator removes the data from the line.

Star Topology

All hosts in Star topology are connected to a central device, known as hub device, using a point-to-point connection. That is, there exists a point-to-point connection between hosts and hub. The hub device can be any of the following:

- Layer-1 device such as hub or repeater
- Layer-2 device such as switch or bridge
- Layer-3 device such as router or gateway

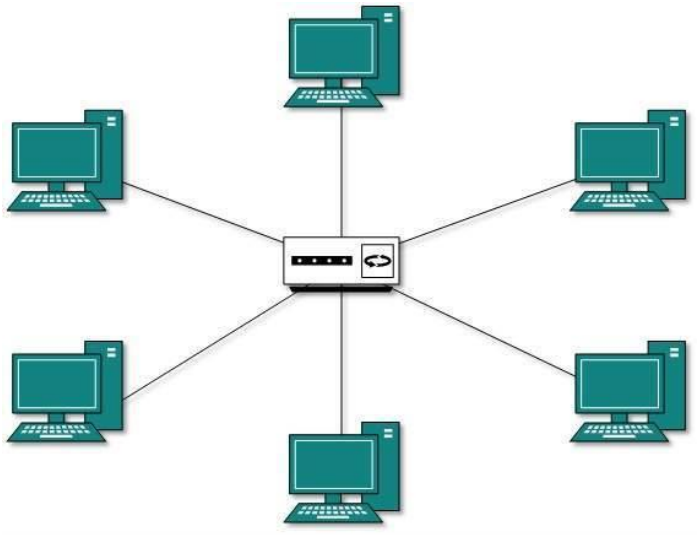


Fig. 3.15 Star Topology

As in Bus topology, [illegible], connectivity of all hosts to all other hosts fails. Every communication between hosts, takes place through only the hub. Star topology is not expensive as to connect one more host, only one cable is required and configuration is simple.

Ring Topology

In ring topology, each host machine connects to exactly two other machines, creating a circular network structure. When one host tries to communicate or send message to a host which is not adjacent to it, the data travels through all intermediate hosts. To connect one more host in the existing structure, the administrator may need only one more extra cable. Ring topology is shown in Figure 3.16.

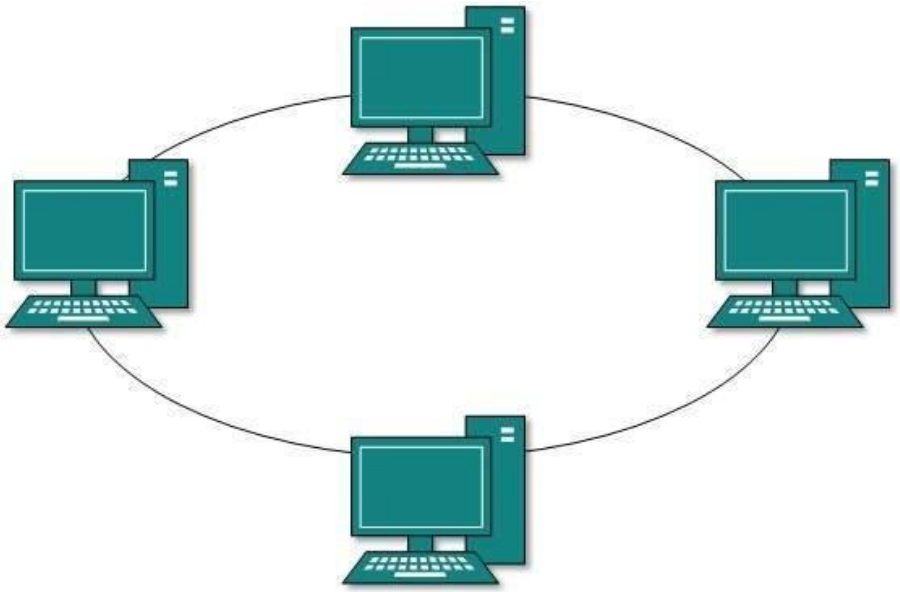


Fig. 3.16 Ring Topology

Failure of any host results in failure of the whole ring. Thus, every connection in the ring is a point of failure. There are methods which employ one more backup ring.

Mesh Topology

In this type of topology, a host is connected to one or multiple hosts. This topology has hosts in point-to-point connection with every other host or may also have hosts which are in point-to-point connection to few hosts only.

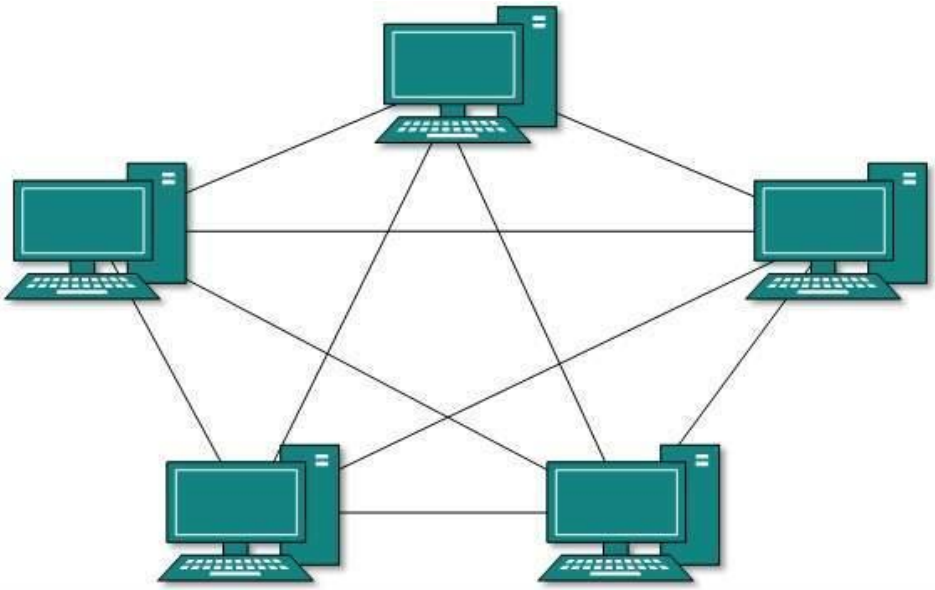


Fig. 3.17 Mesh Topology

Active Network Components vs Passive Network Components

Active Networking Hardware comprises all parts of a computer network which in some way actively work with the signals in the network (amplify them, modify them, evaluate them, etc.). Active Networking Hardware are usually specific devices located in the hubs of the network. Active networking hardware includes:

- Switch
- Repeater
- Hub
- Bridge
- Router
- Print Server
- Access point (AP)

Passive Networking Hardware are those parts of computer network that are involved in data transmission in the network, but they don't change or affect the data. The passive network hardware includes:

- Cables (fiber optic cable, coaxial cables)
- Connectors
- Switchboards

3.5 Configuring a Router

Network configuration is the process of setting a network's controls, flow and operation to support the network communication of an organization and/or network owner. This broad term incorporates multiple configuration and setup processes on network hardware, software and other supporting devices and components.

Before configuring, connect a computer to TP-Link router by cable or through wireless network as shown in Figure 3.18. It is recommended to configure it by a wired connection.

Fig. 3.18 Setting up a Router

Log into the router's web-based utility and follow the following steps:

Step 1

Select **Wireless->Wireless Settings** on the left side menu to open the wireless setting page as shown in Figure 3.19.

Wireless Network Name (also called SSID for some models): Create a new name for your wireless network. If you want to use the default TP-Link_****** wireless name, you can also leave it here as default value.

Region: Select your current location.

"Enable Wireless Router Radio" and **"Enable SSID Broadcast"** should be both ticked.

Please do not change other settings on this page if not necessary.

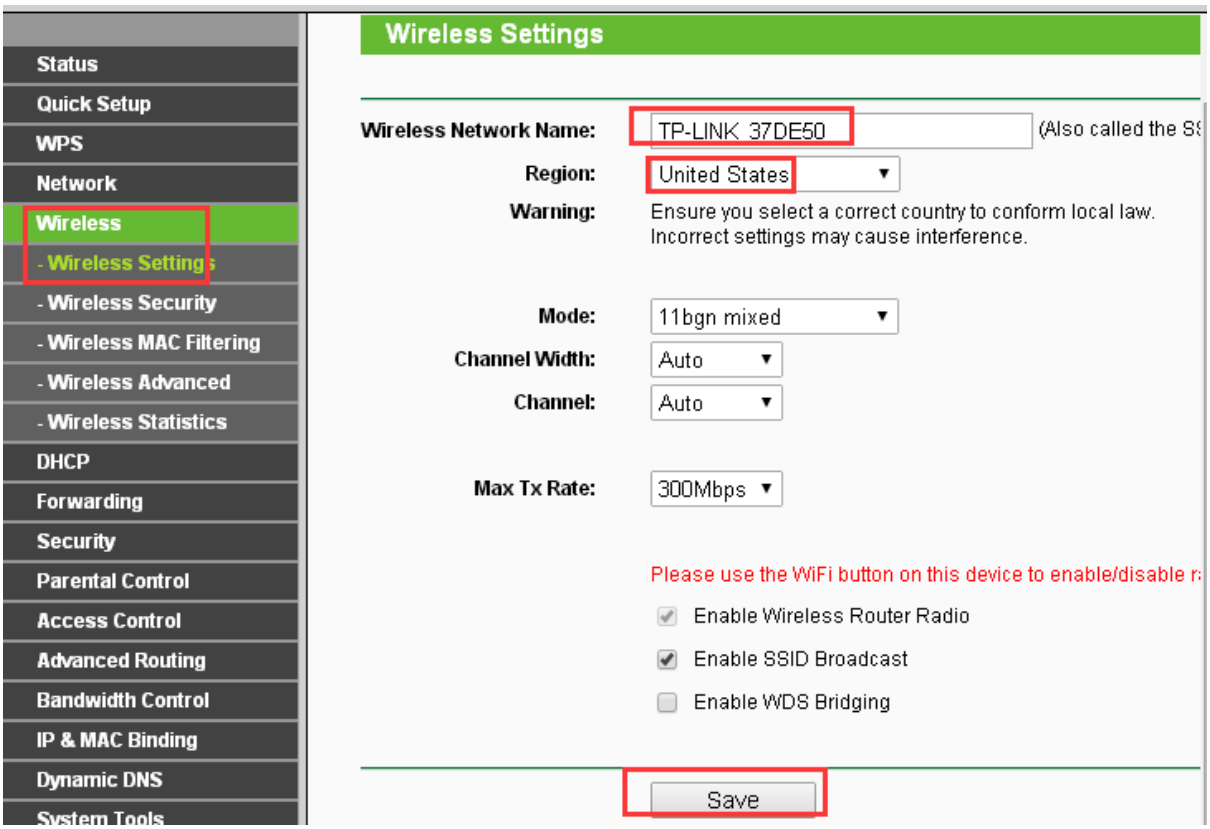


Fig. 3.19 Wireless Settings of a Router

Step 2

Select **Wireless->Wireless Security** on the left side menu as shown in Figure 3.20. It's recommended to use **WPA/WPA2 Personal (Recommended)**, some models will name that as **WPA-PSK/WPA2-PSK** instead.

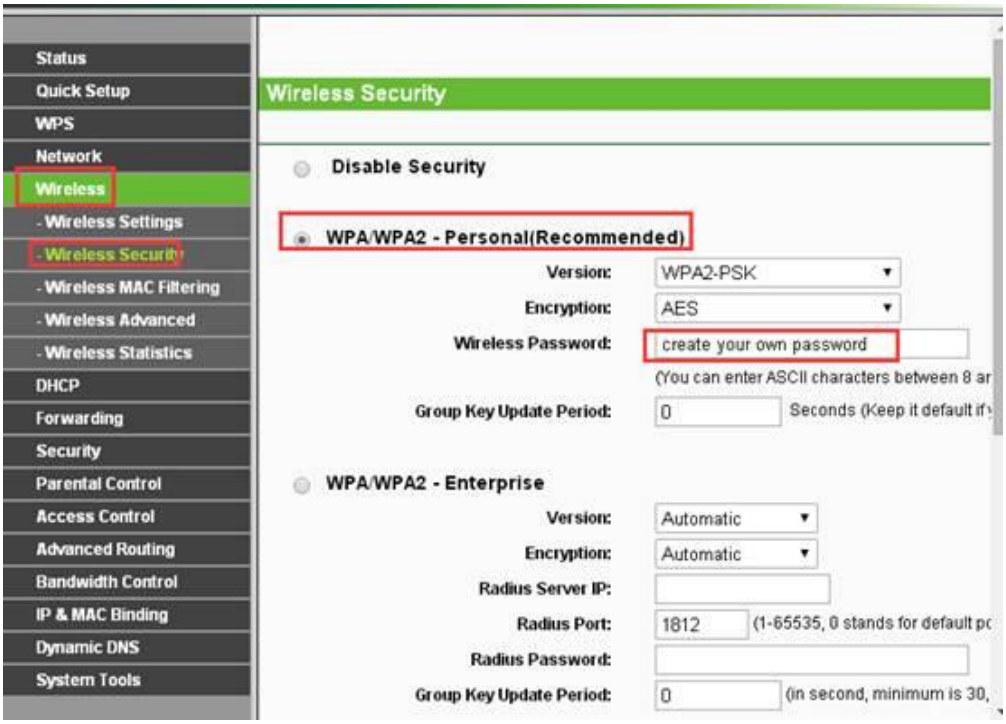


Fig. 3.20 Wireless Security Settings of a Router

Encryption: AES

Wireless Password (also called PSK Password): make up your Wi-Fi password, also called Wireless Network Key.

Setting IPv6

Step 1: Log in to the router's web-based management page.

Step 2: Go to **IPv6 Support** at the left-hand side men as shown in Figure 3.21. In the drop-down menu of **IPv6 Support**, please choose **IPv6 Setup**.

Step 3: Please make sure that **Enable IPv6** has been checked. To Configure **WAN Connection Type**, if you are not sure what the connection type is, please contact your IPv6 provider. Here, take **PPPoEv6** as an example. After the **PPPoEv6** is selected, please input the **User name** and **Password** provided by the IPv6 Provider. For the **Get IPv6 Address Way**, please choose **Get IPv6 prefix delegation**. For the **IPV6 Address Assign Type**, please choose **SLAAC**.

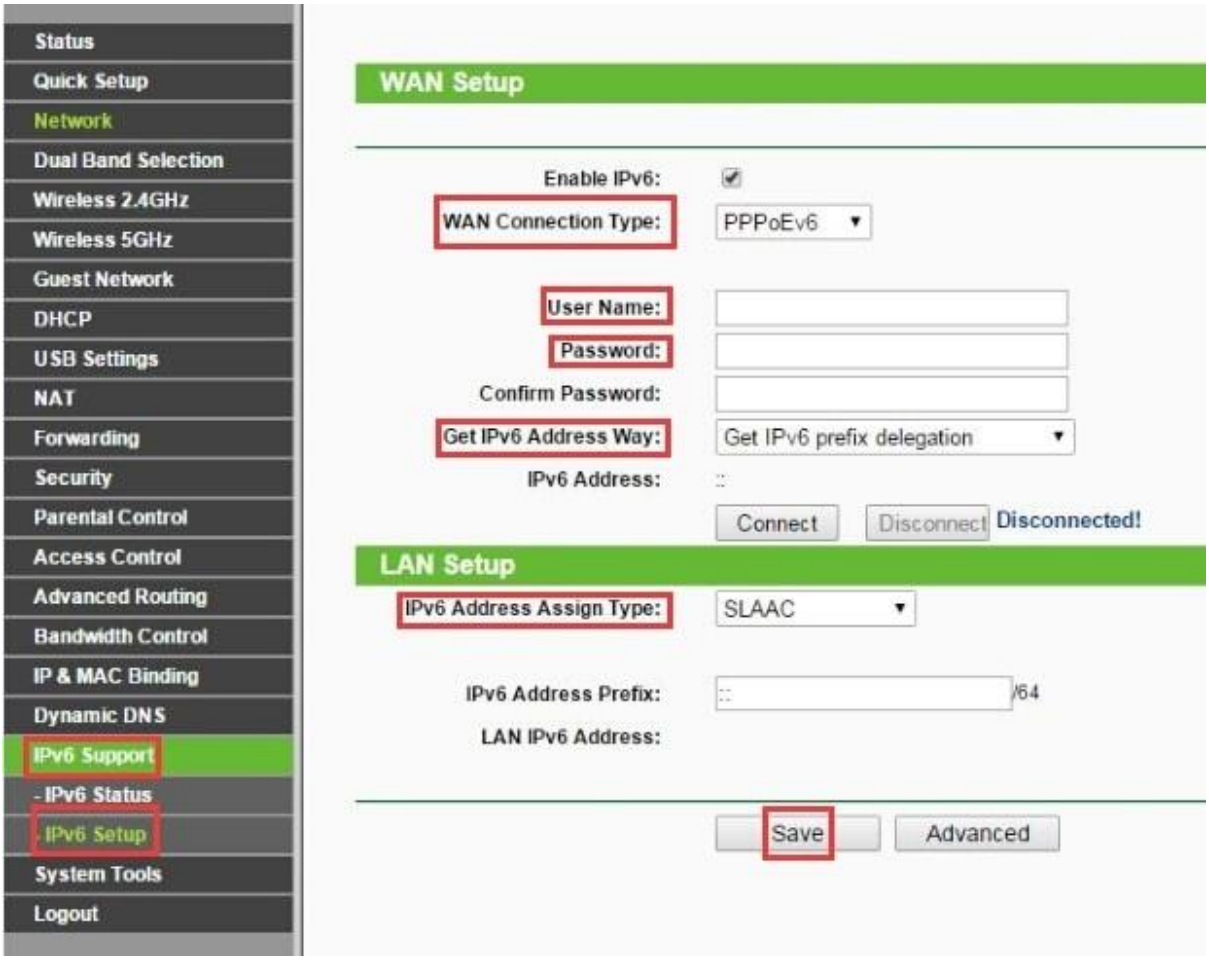


Fig. 3.21 IPv6 Settings of a Router

IP settings of a Network

If your ISP provides the DHCP service, please select *Dynamic IP* (as shown in Figure 3.22), and the router will automatically get IP parameters from your ISP.

Fig. 3.22 IPv6 Settings of a Router

Static IP

If your ISP provides a static or fixed IP address, subnet mask, default gateway and DNS setting, please select Static IP in Figure 3.22 and fill the following fields:

•**IP Address** - Enter the IP address in dotted-decimal notation provided by your ISP.

•**Subnet Mask** - Enter the subnet mask in dotted-decimal notation provided by your ISP. Normally 255.255.255.0 is used as the subnet mask.

•**Gateway** - Enter the gateway IP address in dotted-decimal notation provided by your ISP.

•**Primary/Secondary DNS Server** - (Optional) Enter one or two DNS addresses in dotted-decimal notation provided by your ISP.

•**MTU (Bytes)** - The normal MTU (Maximum Transmission Unit) value for most Ethernet networks is 1500 Bytes. It is not recommended that you change the default MTU size unless required by your ISP.

•**Enable IGMP Proxy** - IGMP (Internet Group Management Protocol) is used to manage multicasting on TCP/IP networks. Some ISPs use IGMP to perform remote

configuration for client devices, such as the modem router. The default value is enabled, and if you are not sure, please contact your ISP or just leave it.

3.6 OSI Model

A typical OSI layered model is shown in Figure 3.23. We'll describe OSI layers "top down" from the application layer that directly serves the end user, down to the physical layer.

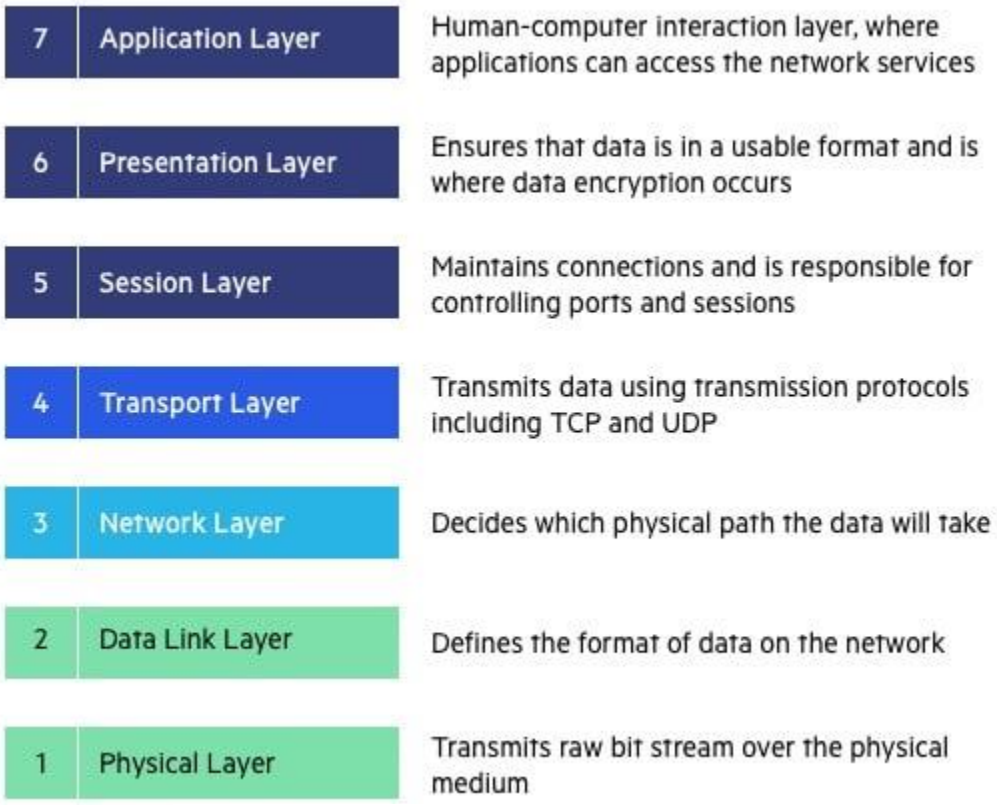


Fig. 3.23 OSI Layered Model

Application Layer: The application layer is used by end-user software such as web browsers and email clients. It provides protocols that allow software to send and receive information and present meaningful data to users. A few examples of application layer protocols are the Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), and Domain Name System (DNS).

Presentation Layer: The presentation layer prepares data for the application layer. It defines how two devices should encode, encrypt, and compress data so it is received correctly on the other end. The presentation layer takes any data transmitted by the application layer and prepares it for transmission over the session layer.

Session Layer: The session layer creates communication channels, called sessions, between devices. It is responsible for opening sessions, ensuring they remain open and functional while data is being transferred, and closing them when communication ends. The session layer can also set checkpoints during a data transfer—if the session is interrupted, devices can resume data transfer from the last checkpoint.

Transport Layer: The transport layer takes data transferred in the session layer and breaks it into "segments" on the transmitting end. It is responsible for reassembling the segments on the receiving end, turning it back into data that can be used by the session layer. The transport layer carries out flow control, sending data at a rate that matches the connection speed of the receiving device, and error control, checking if data was received incorrectly and if not, requesting it again.

Network Layer: The network layer has two main functions. One is breaking up segments into network packets, and reassembling the packets on the receiving end. The other is routing packets by discovering the best path across a physical network. The network layer uses network addresses (typically Internet Protocol addresses) to route packets to a destination node.

Data Link Layer: The data link layer establishes and terminates a connection between two physically-connected nodes on a network. It breaks up packets into frames and sends them from source to destination. This layer is composed of two parts—Logical Link Control (LLC), which identifies network protocols, performs error checking and synchronizes frames, and Media Access Control (MAC) which uses MAC addresses to connect devices and define permissions to transmit and receive data.

Physical Layer: The physical layer is responsible for the physical cable or wireless connection between network nodes. It defines the connector, the electrical cable or

wireless technology connecting the devices, and is responsible for transmission of the raw data, which is simply a series of 0s and 1s, while taking care of bit rate control.

TCP/IP model

The TCP/IP model was developed prior to the OSI model. The TCP/IP model is not exactly similar to the OSI model.

- The TCP/IP model consists of five layers: the application layer, transport layer, network layer, data link layer and physical layer.
- The first four layers provide physical standards, network interface, internetworking, and transport functions that correspond to the first four layers of the OSI model and these four layers are represented in TCP/IP model by a single layer called the application layer.
- TCP/IP is a hierarchical protocol made up of interactive modules, and each of them provides specific functionality.

Here, hierarchical means that each upper-layer protocol is supported by two or more lower-level protocols.

OSI vs. TCP/IP Model

The Transfer Control Protocol/Internet Protocol (TCP/IP) is older than the OSI model and was created by the US Department of Defence (DoD). A key difference between the models is that TCP/IP is simpler, collapsing several OSI layers into one:

- OSI layers 5, 6, 7 are combined into one Application Layer in TCP/IP as shown in Figure 3.24.
- OSI layers 1, 2 are combined into one Network Access Layer in TCP/IP – however TCP/IP does not take responsibility for sequencing and acknowledgement functions, leaving these to the underlying transport layer.

- TCP/IP is a functional model designed to solve specific communication problems, and which is based on specific, standard protocols. OSI is a generic, protocol-independent model intended to describe all forms of network communication.

- In TCP/IP, most applications use all the layers, while in OSI simple applications do not use all seven layers. Only layers 1, 2 and 3 are mandatory to enable any data communication.

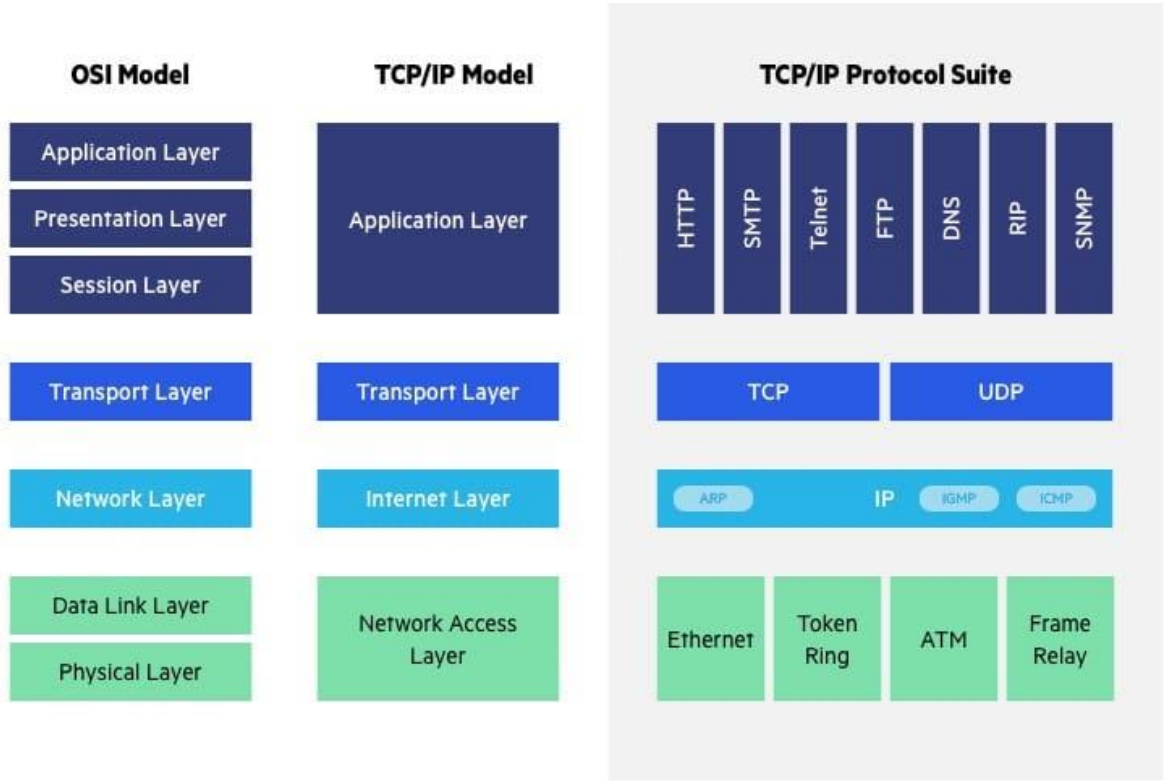


Fig. 3.24 OSI vs TCP/IP Model

What is IP?

An **IP (Internet Protocol) address** is a numerical label assigned to each device connected to a computer network that uses the IP protocol for communication. An IP address acts as an identifier for a specific device on a particular network. The IP address is also called an IP number or Internet address. IP address specifies the technical format of the addressing and packets scheme. Most networks combine IP with a TCP (Transmission Control Protocol). It also allows developing a virtual connection between a destination and a source.

IPv4

IPv4 is an IP version widely used to identify devices on a network using an addressing system. It was the first version of IP deployed for production in the ARPANET in 1983. It uses a 32-bit address scheme to store 2^32 addresses which is more than 4 billion addresses. It is considered the primary Internet Protocol and carries 94% of Internet traffic.

IPv6

IPv6 is the most recent version of the Internet Protocol. This new IP address version is being deployed to fulfill the need for more Internet addresses. It was aimed to resolve issues that are associated with IPv4. With 128-bit address space, it allows 340 undecillion unique address space.

Static IP address:

A static IP address is simply an address that doesn't change. Once your device is assigned a static IP address, that number typically stays the same until the device is decommissioned or your network architecture changes. Static IP addresses generally are used by servers or other important equipment.Static IP addresses are assigned by Internet Service Providers (ISPs). Your ISP may or may not allocate you a static IP address depending on the nature of your service agreement.

Dynamic IP address

As the name suggests, dynamic IP addresses are subject to change, sometimes at a moment's notice. Dynamic addresses are assigned, as needed, by Dynamic Host Configuration Protocol (DHCP) servers.We use dynamic addresses because IPv4 doesn't provide enough static IP addresses to go around. So, for example, a hotel probably has a static IP address, but each individual device within its rooms would have a dynamic IP address.On the internet, your home or office may be assigned a dynamic IP address by your ISP's DHCP server.

MAC Address

The MAC address is a unique value associated with a network adapter. MAC addresses are also known as **hardware** addresses or **physical** addresses. They uniquely identify an adapter on a LAN. MAC addresses are 12-digit hexadecimal numbers (48 bits in length). By convention, MAC addresses are usually written in one of the following formats:

MM:MM:MM:SS:SS:SS

MMMM-MMSS-SSSS

Subnet mask:

A 32-bit combination used to describe which portion of an address refers to the subnet and which part refers to the host.

Classes of IP addresses

TCP/IP defines five classes of IP addresses: class A, B, C, D, and E. Each class has a range of valid IP addresses. The value of the first octet determines the class. IP addresses from the first three classes (A, B and C) can be used for host addresses. The other two classes are used for other purposes – class D for multicast and class E for experimental purposes.

3.7 Testing a Network

The basic ping test is one of the easiest things you can do to verify connectivity between your computer and a remote host. This test is easy to run from the command line.

Using ping test on Microsoft Windows

To use the ping program on Microsoft Windows, follow these steps:

1. Open a DOS command window. To do this, click **Start**, click **Run**, type *cmd*, and then press Enter.
2. At the command prompt, type the following command. Replace *example.com* with the domain that you want to test:
   Ping example.com
3. Interpret the output from ping:

1. If the remote host is active and configured to respond to ping requests, responses appear. For example, the following output shows ping responses from an A2 Hosting server:

```
C:\Documents and Settings\user>ping a2s78.a2hosting.com
```

- Pinging a2s78.a2hosting.com [216.119.143.98] with 32 bytes of data:
- Reply from 216.119.143.98: bytes=32 time=46ms TTL=54
- Reply from 216.119.143.98: bytes=32 time=45ms TTL=54
- Reply from 216.119.143.98: bytes=32 time=47ms TTL=54
- Ping statistics for 216.119.143.98:
- Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
- Approximate round-trip times in milli-seconds:

```
Minimum = 45ms, Maximum = 47ms, Average = 46ms
```

Alternatively, if the remote host is down, or not configured to respond to ping requests, you do not see any responses.

Using Router Interface

To check internet connectivity using router interface, follow the steps given below:

1. Open Router configuration page.

2. Click System Tools and Diagnostic.

3. Select one of the diagnostic tools: Ping.

Note: Ping troubleshoots connectivity, reachability, and name resolution to a given host or gateway.

4. Type the IP Address/Domain name of the PC/server you want to ping the IP Address/ Domain Name, as shown in Figure 3.25.

5. Start Diagnose and analyze the results.

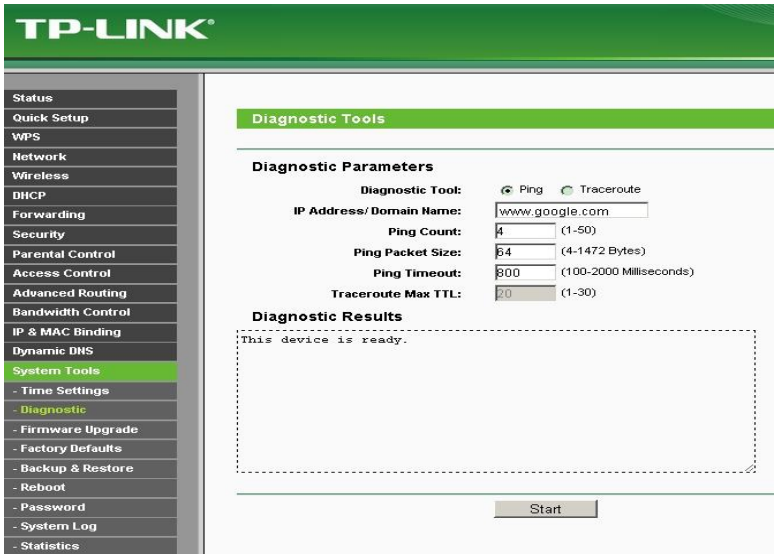


Fig. 3.25 Router Interface: Checking Internet

7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium

Key Points

- A computer network is a group of computers linked to each other that enables the computer to communicate with another computer and share their resources, data, and applications.
- Local Area Network is a group of computers connected to each other (as shown in Figure 3.2) in a small area such as building, office.
- Personal Area Network is a network arranged within an individual person, typically within a range of 10 meters.
- A metropolitan area network is a network that covers a larger geographic area by interconnecting a different LAN to form a larger network.
- A Wide Area Network is a network that extends over a large geographical area such as states or countries.
- Hub is a device that splits a network connection into multiple computers.
- Switch uses physical device addresses in each incoming messages so that it can deliver the message to the right destination or port.
- The device used to connect a LAN with an internet connection is called Router.
- An Ethernet cable is a network cable used for high-speed wired network connections between two devices.
- The speed of the data is expressed in bits per second (bits/s or bps).
- Baud rate refers to the number of signal or symbol changes that occur per second.
- Attenuation in computer networking is the loss of communication signal strength that is measured in decibels (dB).
- When you are accessing a wireless hotspot, or the router in your home or office, your device is the client.
- Access points are devices that host and control the wireless connection for laptops, tablets, or smart phones.
- Mac Address is the physical address of device.

Exercise

Select the most appropriate option

1. Which of the following is the hardware/physical address of a computer?
   a. IP Address b. MAC Address c. Static IP d. Subnet Mask
2. Speed of data is expressed in:
   a. Hz/s b. words/s c. bits/s d. bits/min
3. Which of the following is not an active network component?
   a. Cable b. Router c. Hub d. Switch
4. Ping is used to
   a. Assign IP b. Login c. Test a network d. search files
5. There are ____ classes of IP addresses.
   a. 4 b. 3 c. 5 d. None of these

Write short answer of the following.

1. Define Computer Networks.
2. Differentiate between wired and wireless networks.
3. What is the difference between accesspoint and stations?
4. Define Router.
5. Define Switch.

Answer the following question in detail.

1. Differentiate between LAN and PAN.
2. Differentiate between IP address and MAC address.
3. Differentiate between IPv6 and IPv4.
4. Describe network topologies.
5. Describe DHCP mode.

Practical Tasks

1. Assign static IP addresses to the computers in the computer lab.
2. Perform Ping test to check the Internet connectivity of your computer.

Chapter 4

Basics of Programming Language

After Studying this chapter, you will be able to

- describe computer languages.
- explain types of languages: low level language; high level language.
- use of different languages for different applications.
- define translational software.
- understand types of translational software: Assembler, Interpreter and Compiler.
  understand computer programming cycle: coding, translation, execution and debugging.
- explain IDE.
- define flow chart.
- understand symbols of flow chart.
  design flow diagram for a simple problem (addition, subtraction, multiplication, division, and condition (if else)).

4.1 PROGRAMMING

Programming is talking to computer. It is a language to communicate with your computer. The main unit of a computer is microprocessor. Each microprocessor is designed to execute a certain set of instructions. Through programming, we pass these instructions to computer and computer executes these instructions in given order.

A Computer language is defined as a code or a syntax which is used to write programs or any specific applications. Computer language is used to communicate with computers. Broadly, it can be classified into three categories assembly language, machine language, and high-level language. The machine language is considered as oldest computer language among all three. In machine language, the input is directly given as binary input which is processed by the machine. Binary inputs mean one and zero form. For computer language processing the system needs compiler and interpreter to convert the language in computer language so that it can be processed by a machine.

What is inside a Microprocessor?

A microprocessor contains millions of transistors and capacitive elements. All these components operate after providing voltages. Presence of voltage means a signal 1 and absence of voltage means signal 0. So, a computer is only able to understand presence and absence of voltages.

Machine Language to C

Computer only understands machine language also called binary language. A machine language instruction consists of only 0s and 1s. For example, to add two numbers, machine language instruction can be 00011001. It is very difficult to remember and understand these machine language instructions for programmers.

To make the instructions understandable to the programmer, assembly language was formed. Assembly language consists of some simple statements i.e., sum a, b. This instruction is supposed to add a and b. Each processor has its own assembly language instruction set. So, it can only execute instructions from its instruction set. Whenever an assembly language instruction is executed, it is first converted into equivalent machine

language instruction (consisting of 0s and 1s).

Although assembly language is easier to understand for the programmers, there is still missing information in assembly language instructions, for example, *sum a, b* is equivalent to a = a + b, but it cannot be inferred from the instruction that result will be assigned to *a*. To facilitate the programmers further, another language was designed named "C Language". This language was implemented to write programs in mathematical language. In C language, if we have to sum a and b and store the result in a, the instruction will be: $a = a + b$. There is no hidden information in this statement.

What is Programming?

When a C program is executed, it is first translated into equivalent assembly language code, which is further translated into equivalent machine language code. The computer finally executes machine language instructions being translated from C program. Hence a C program is "a **set of sequence of commands** being sent to microprocessor for execution."

Programming is not only writing instructions to the computer but it is instructions in the right sequence to form a required logic (i.e., to sum a and b etc.).

Integrated Development Environment for C

An IDE, or Integrated Development Environment, enables programmers to consolidate the different aspects of writing a computer program. IDEs increase programmer productivity by combining common activities of writing software into a single application: editing source code, building executables, and debugging.

Compilation of a C program converts C code to machine language instructions. There are many softwares available for compiling and running C programs. Some famous IDEs are Microsoft Visual Studio, Visual C++ and Eclipse etc. Online compilers can also be explored. In this chapter, we will only learn to use Microsoft Visual Studio.

Do you know?

There is a step wise debugger available in many IDEs which helps you to troubleshoot your program.

Point to Ponder

- What is the difference between C and C++ languages?

4.2 Different Types of Computer Languages:

Below are the top 3 types of computer language:

1. Machine Language

The machine language is sometimes referred to as machine code or object code which is a set of binary digits: 0 and 1. These binary digits are understood and read by a computer system and it interprets it easily. It is considered a native language as it can be directly understood by a central processing unit (CPU). The machine language is not so easy to understand, as the language uses the binary system in which the commands are written in 1 and 0 form which is not easy to interpret. There is only one language which is understood by computer language which is machine language. The operating system of the computer system is used to identify the exact machine language used for that particular system.

The operating system defines how the program should be written so that it can be converted to machine language and the system takes appropriate action. The computer programs and scripts can also be written in other programming languages like C, C++, and JAVA. However, these languages cannot be directly understood by a computer system so there is a need for a program that can convert these computer programs to machine language. The compiler is used to convert the programs to machine language which can be easily understood by computer systems. The compiler generates the binary file and executable file.

Example of machine language for the text “Hello World”.

01001000 0110101 01101100 01101100 01101111 00100000 01010111 01101111 01110010 01101100 01100100.

2. Assembly Language

The assembly language is considered a low-level language for microprocessors and many other programmable devices. The assembly language is also considered as second-generation language. The first-generation language is machine language. The assembly language is mostly famous for writing an operating system and also in writing different desktop applications. The operations carried out by programmers using assembly language are memory management, registry access, and clock cycle operations. The drawback of assembly language is that the code cannot be reused and the language is not so easy to understand. The assembly language is considered a group of other languages. It is used to implement the symbolic representation of machine code which is used to program CPU architecture. The other name of assembly language is assembly code. For any processor, the most used programming language is assembly language.

In assembly language, the programmer does the operation which can be directly executed on a central processing unit (CPU). The language has certain drawbacks as it does not contain any variables or functions in programs and also the program is not portable on different processors. The assembly language uses the same structure and commands which machine language does use but it uses names in place of numbers. The operations performed using the assembly language is very fast. The operations are much faster when it is compared to high-level language.

Activity

- Find *sum* command for different microcontrollers from their datasheets.

Do you know

- Every microprocessor has its own assembly language instructions.

3. High-Level Languages

The code written in low-level language for one machine cannot be transferred to other machines. Thus, it led to the development of high-level language. The high-level language is easy to understand as the programs written are user-friendly. The other advantage of code written in a high-level language is that the code is independent of a computer system which means the code can be transferred to other machines. The high-level of language uses the concept of abstraction and also focus on programming language rather than focusing on computer hardware components like register utilization or memory utilization.

Higher-level languageswere created to write a human-readable program that can be easily understood by any user. The only requirement in a high-level language is the need of compiler. The program written in a high-level language is not directly understood by the computer. It needs to be converted to machine level language before the execution of high-level programs,. The examples of high-level language are C++, C, JAVA, FORTRAN and Python etc.

When you are programming to control the behavior of hardware components, then low level programming languages are used. For Instance, C is used to interface sensors with microcontrollers. Whereas, Python is used for Artificial Intelligence.

4.3 Language Translators

A translator is a programming language processor that converts a computer program from one language to another. It takes a program written in source code and converts it into machine code. It discovers and identifies the error during translation.

Purpose of Translator

It translates high-level language program into a machine language program that the central processing unit (CPU) can understand. It also detects errors in the program.

Roles of translator are

• Translating the high-level language program input into an equivalent machine language program.

• Providing diagnostic messages wherever the programmer violates specification of the high-level language program.

Different Types of Translators

There are 3 different types of translators as follows:

Compiler

A compiler is a translator used to convert high-level programming language to low-level programming language. It converts the whole program in one session and reports errors detected after the conversion. Compiler takes time to do its work as it translates high-level code to lower-level code all at once and then saves it to memory.

A compiler is processor-dependent and platform-dependent. This dependence has been addressed by a special compiler, a cross-compiler (which compiles a program for another different microprocessor) and a source-to-source compiler. Before choosing a compiler, user has to identify first the Instruction Set Architecture (ISA), the operating system (OS) and the programming language that will be used to ensure that it will be compatible.

Interpreter

Just like a compiler, interpreter is a translatorused to convert high-level programming language to low-level programming language. It converts the program one at a time and reports errors detected at once, while doing the conversion. With this, it is easier to detect errors than in a compiler. An interpreter is faster than a compiler as it immediately executes the code upon reading the code.

It is often used as a debugging tool for software development as it can execute a single line of code at a time. An interpreter is also more portable than a compiler as it is not processor-dependent, you can work between hardware architectures.

Assembler

An assembler is a translator used to translate assembly language to machine language. It is like a compiler for the assembly language but interactive like an interpreter. Assembly language is difficult to understand as it is a low-level programming language. An assembler translates a low-level language, an assembly language to an even lower-level language, which is the machine code. The machine code can be directly understood by the CPU.

4.4 Program Development Life Cycle

When we develop a program using any programming language, we follow a sequence of steps. These steps are called phases in program development. The program development life cycle is a set of steps or phases that are used to develop a program in any programming language.

Generally, the program development life cycle contains 6 phases as shown in Figure 4.1, they are as follows:

- Problem Definition
- Problem Analysis
- Algorithm Development
- Coding & Documentation
- Testing & Debugging
- Maintenance

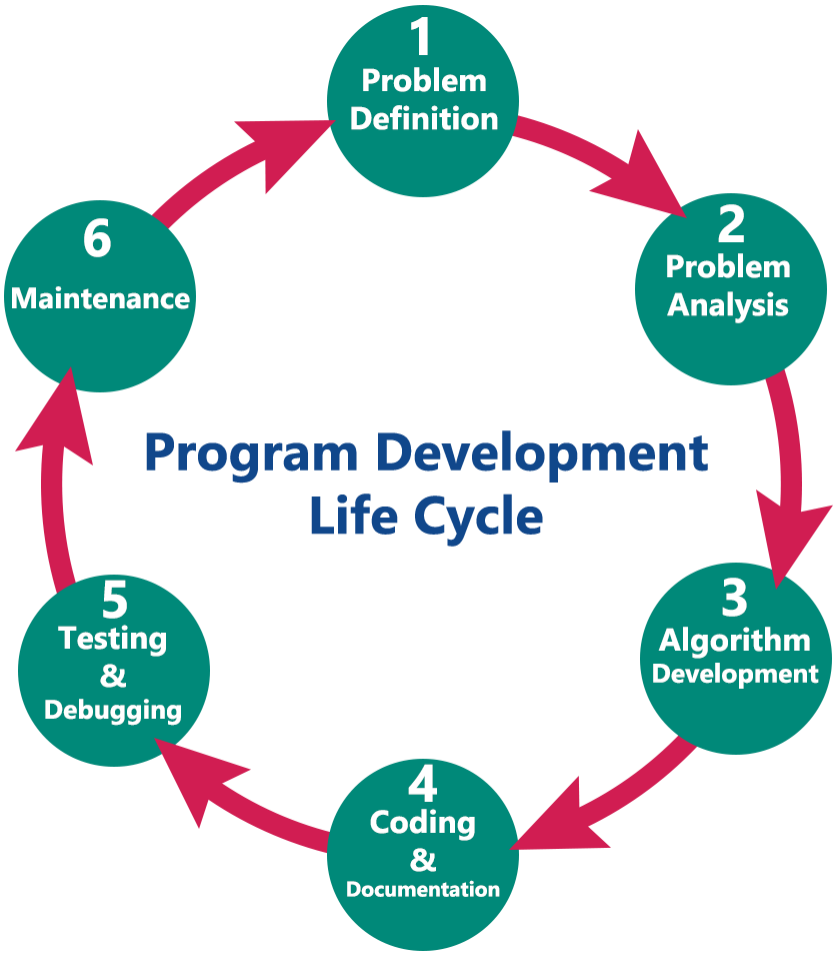


Fig. 4.1 Program Development Life Cycle

1. Problem Definition

In this phase, we define the problem statement and we decide the boundaries of the problem. In this phase we need to understand the problem statement, what is our requirement, what should be the output of the problem solution. These are defined in this first phase of the program development life cycle.

2. Problem Analysis

In phase 2, we determine the requirements like variables, functions, etc. to solve the problem. That means we gather the required resources to solve the problem defined in the problem definition phase. We also determine the bounds of the solution.

3. Algorithm Development

During this phase, we develop a step-by-step procedure to solve the problem using the specification given in the previous phase. This phase is very important for program development.

4. Coding & Documentation

This phase uses a programming language to write or implement the actual programming instructions for the steps defined in the previous phase. In this phase, we construct the actual program. That means we write the program to solve the given problem using programming languages like C, C++, Java, etc.,

5. Testing & Debugging

During this phase, we check whether the code written in the previous step is solving the specified problem or not.We also test whether it is providing the desired output or not.

6. Maintenance

During this phase, the program is actively used by the users. If the user encounters any problem or wants any enhancement, then we need to repeat all the phases from the starting, so that the encountered problem is solved or enhancement is added.

4.5 Flow Charts

A flowchart is simply a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes. Typically, a flowchart shows the order of steps by connecting them with arrows. Figure 4.2 shows an example flowchart.

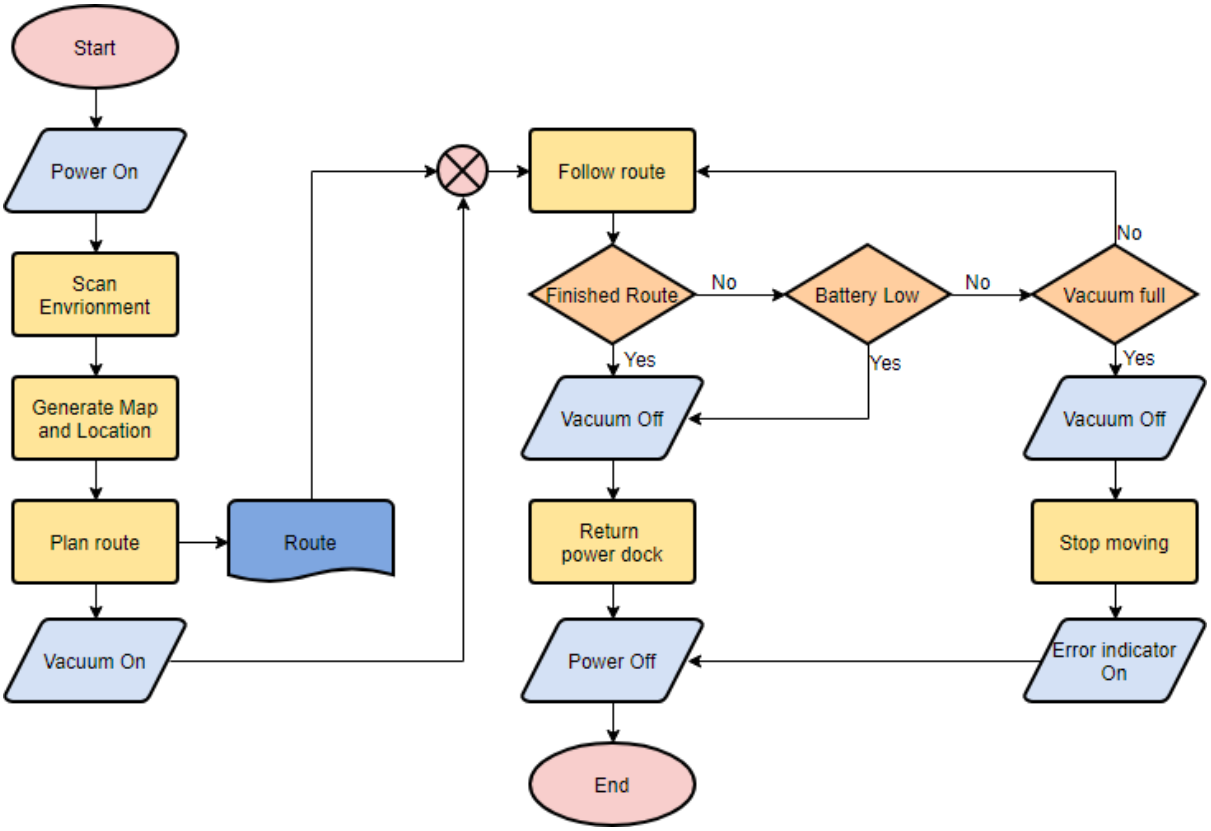


Fig. 4.2 Flow Chart Example

A flowchart is a graphical representation of steps. It was originated from computer science as a tool for representing algorithms and programming logic but had extended to use in all other kinds of processes. Nowadays, flowcharts play an extremely important role in displaying information and assisting reasoning. They help us visualize complex processes, or make explicit the structure of problems and tasks. A flowchart can also be used to define a process or project to be implemented.

4.5.1 Flow chart Symbols

Different flowchart shapes have different conventional meanings. The meanings of some of the more common shapes are as follows:

Terminator

The terminator symbol represents the starting or ending point of the system.

Process

A box indicates some particular operation.

Document

This represents a printout, such as a document or a report.

Decision

A diamond represents a decision or branching point. Lines coming out from the diamond indicates different possible situations, leading to different sub-processes.

Data

It represents information entering or leaving the system. An input might be an order from a customer. Output can be a product to be delivered.

On-Page Reference

This symbol would contain a letter inside. It indicates that the flow continues on a matching symbol containing the same letter somewhere else on the same page.

Off-Page Reference

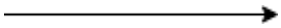
This symbol would contain a letter inside. It indicates that the flow continues on a matching symbol containing the same letter somewhere else on a different page.

Delay

Identifies a delay.

Flow

Lines represent the flow of the sequence and direction of a process.



Benefits of Flowcharts

Using a flowchart has a variety of benefits:

- It helps to clarify complex processes.
- It identifies steps that do not add value to the internal or external customer, including delays; needless storage and transportation; unnecessary work, duplication, and added expense; breakdowns in communication.
- It helps team members gain a shared understanding of the process and use this knowledge to collect data, identify problems, focus discussions, and identify resources.
- It serves as a basis for designing new processes.

4.5.2 Flowchart Examples

In this section, we present some flowchart examples.

Medical Service

This is a hospital flowchart example that shows how clinical cases shall be processed. This flowchart (Figure 4.3) uses decision shapes intensively in representing alternative flows.

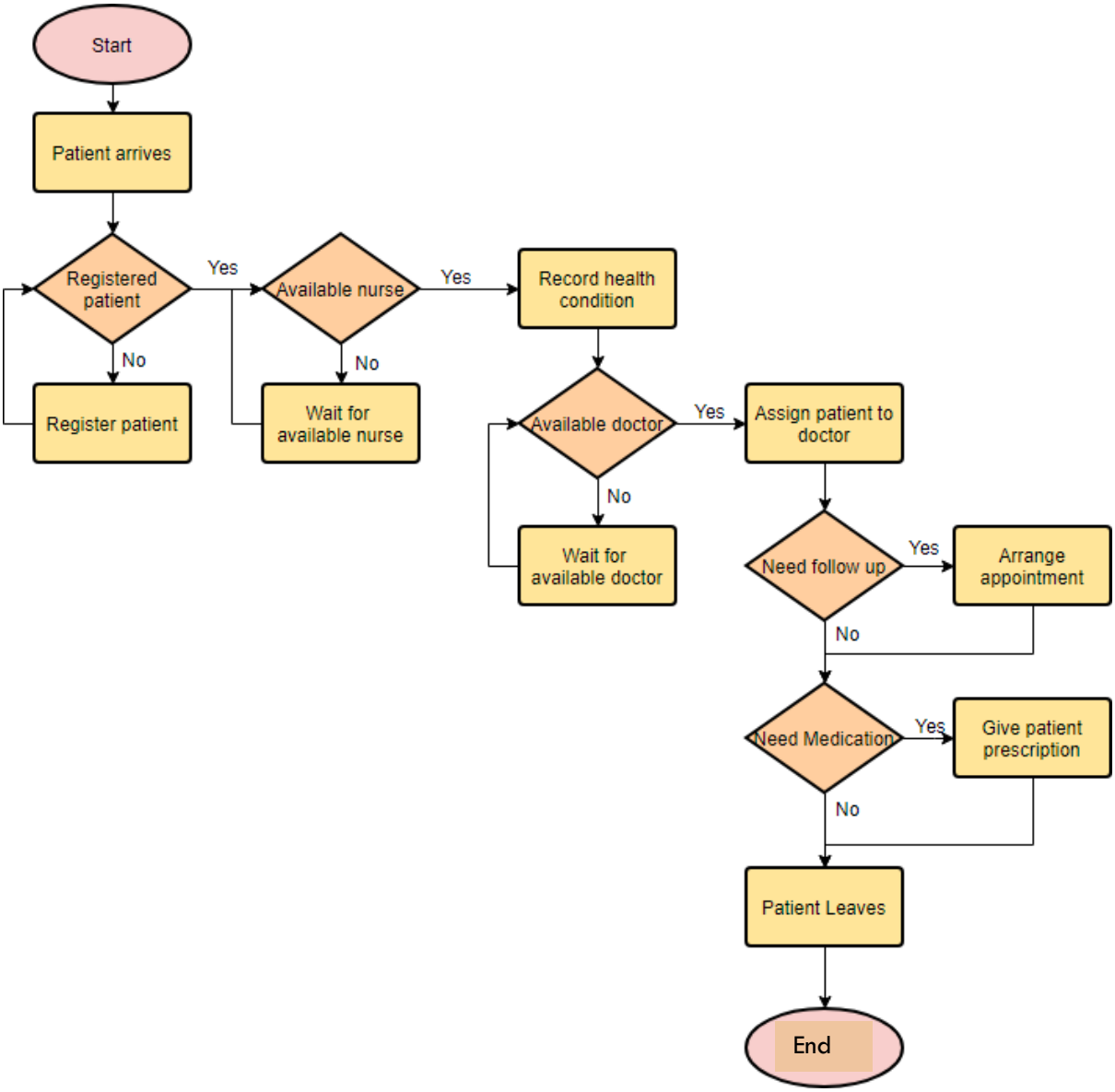


Fig. 4.3 Flow Chart Example: Medical

Simple Algorithms

A flowchart can also be used in visualizing algorithms, regardless of its complexity. Here is an example (Figure 4.4) that shows how flowchart can be used in showing a simple summation process.

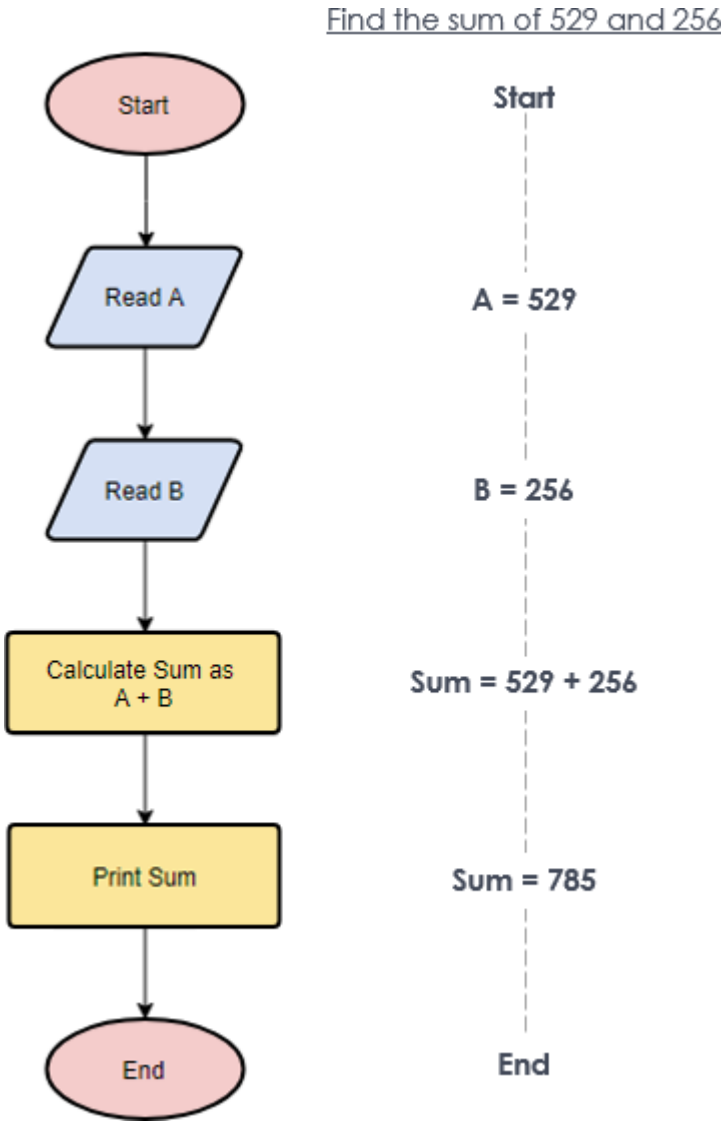


Fig. 4.4 Flow Chart Example: Sum

Calculate Profit and Loss

The flowchart example below (Figure 4.5) shows how profit and loss can be calculated.

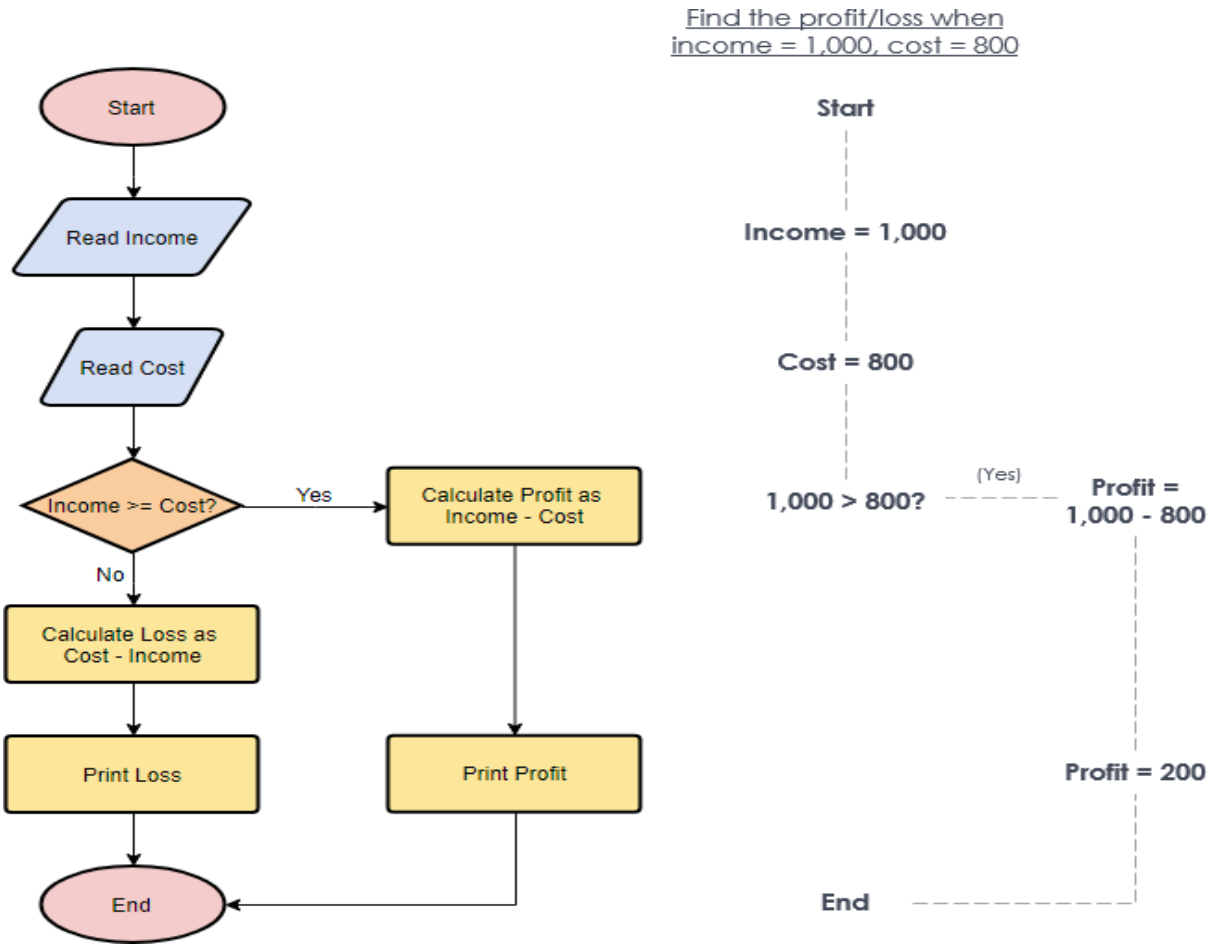


Fig. 4.4 Flow Chart Example: Profit and Loss Calculation

Note for Teachers.

- Give examples of recursive flow charts.

Key Points

- The computer language is defined as code or syntax which is used to write programs or any specific applications.
- The main unit of a computer is microprocessor. Each microprocessor is designed to execute a certain set of instructions.
- Presence of voltage means a signal 1 and absence of voltage means signal 0. So, a computer is only able to understand presence and absence of voltages.
- Computer only understands machine language also called binary language.
- Assemble language consists of some simple statements i.e., sum a, b.
- C language was implemented to write programs in mathematical language.
- Hence a C program is "a **set of sequence of commands** being sent to microprocessor for execution."
- An IDE, or Integrated Development Environment, enables programmers to consolidate the different aspects of writing a computer program.
- There is a step wise debugger available in many IDEs which helps you to troubleshoot your program.
- The other advantage of code written in a high-level language is the code is independent of a computer system which means the code can be transferred to other machines.
- A translator is a programming language processor that converts a computer program from one language to another.
- An assembler is a translator used to translate assembly language to machine language

Exercise

Select the most appropriate option

1. Assembly is a

   a. High level language b. Low level language c. Machine Language d. Both a and b

2. Which of the following language is different for all microcontrollers.

a. Assembly Language b. Machine Language c. C Language d. Both a and

3. _____ converts the assembly language code to machine language.

   a. Compiler b. Assembler c. Linker d. None of these

4. Which of the following language supports mathematical expressions.

   a. C b. Assembly c. Machine Language d. none of these

5. ____ are used to understand flow of a computer algorithm.

   a. Flow charts b. Line graphs c. Bar charts d. none of these

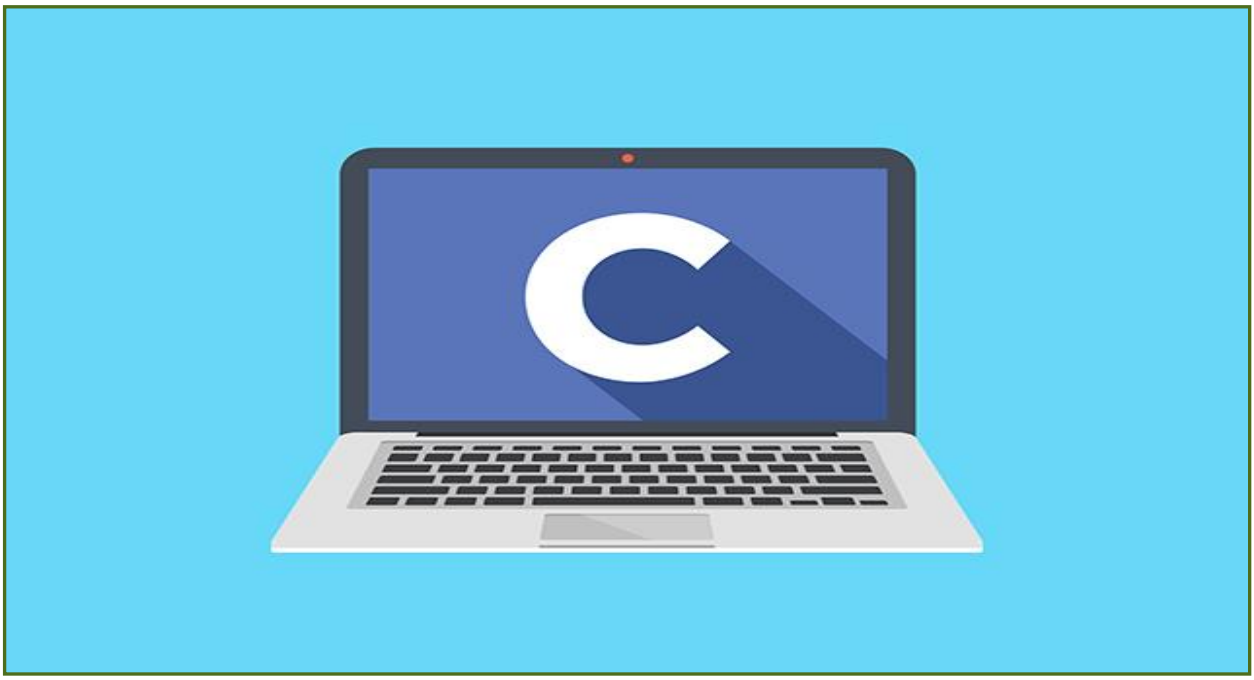
write short answer of the following.

1. What is programming?
2. What is the difference between high level and low level programming languages?
3. Define program development cycle.
4. What is the difference between a debugger and a compiler.

Answer the following question in detail.

1. Explain program translation steps?
2. What are flow chart and why are they used?
3. Differentiate between Interpreter and Compiler.

Chapter no. 5
Introductions to C



After Studying this chapter, you will be able to

- understand structure and life cycle of C program.
- define integrated development environment (IDE).
- install C Compiler and IDE.
- develop a program of printing hello world in Programming language C/C++.
- understand input and output statements.
- understand of format specifier for print and scan statement.
- understand data types and variable.
- understand mathematical operators available in C.
- understand precedence of operators.
- implement arithmetic expression in C.
- describe and implement conditional statements and its types.
- understand nested decision statements.
- define concept of loop.
- understand handling control variables for a loop.
- understand breaking a loop.
- implement loops in a program in C.
- understand nested loop.
- describe functions and its types.
- understand return types and input parameters.
- describe data types of arrays and its declaration.
- understand “string”.

5.1 C Programming

C is a general-purpose, procedural computer programming language supporting structured programming, and recursion, with a static type system. C is the most widely used computer language. It keeps fluctuating at number one scale of popularity along with Java programming language and Python scripting.

Interesting Information:

C programming is a general-purpose, procedural, imperative computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system.

Why to Learn C Programming?

C programming language is a must for students and working professionals to become a software engineer especially when they are working in software development domain. Some key advantages of learning C Programming are:

- Easy to learn.
- Structured language.
- It produces efficient programs.
- It can handle low-level activities.
- It can be compiled on a variety of computer platforms.

Facts about C

- C was invented to write an operating system called UNIX.
- C is a successor of B language which was introduced around the early 1970s.
- The language was formalized in 1988 by the American National Standard Institute (ANSI).
- The UNIX OS was totally written in C.
- Today C is the most widely used and popular System Programming Language.
- Most of the state-of-the-art software have been implemented using C.

5.2 Structure of a C Program

Hello World Example

A C program basically consists of the following parts:

- Preprocessor Commands
- Statements & Expressions
- Main Function

As an example, see the following program to print “Hello World”:

```
#include<stdio.h>    // Preprocessor Commands

int main(){          // Main Function
/* my first program in C */
   printf("Hello, World! \n");  // Statements

return0;
}
```

- The first line of the program *#include <stdio.h>* is a preprocessor command, which tells a C compiler to include *stdio.h* file before going to actual compilation.
- The next line *int main()* is the main function where the program execution begins.
- The next line /*...*/ will be ignored by the compiler and it has been put to add additional comments in the program. Such lines are called comments in the program.
- The next line *printf(...)* is another function available in C which causes the message "Hello, World!" to be displayed on the screen.
- The next line **return 0;** terminates the main() function and returns the value 0.

5.3 Life Cycle of a C Program

A C program goes through the following stages before execution:

- **Preprocessing** - Using a Preprocessor program to convert C source code in expanded source code. "#includes" and "#defines" statements will be processed and replaced with the source codes in this step.

- **Compilation** - Using a Compiler program to convert C expanded source to assembly source code.
- **Assembly** - Using an Assembler program to convert assembly source code to object code.
- **Linking** - Using a Linker program to convert object code to executable code. Multiple units of object codes are linked together in this step.
- **Loading** - Using a Loader program to load the executable code into CPU for execution.

5.4 Setting up Programming Environment

There are plenty of Integrated Development Environments (IDEs) available for C language. We will only focus of Microsoft Visual Studio in this book.

Getting Started with Microsoft Visual Studio

1. Download the setup from official visual studio website.
2. Install visual studio on your computer.
3. Open the application
4. Click new project
5. Select Empty Project and name the project as shown in Figure 5.1.

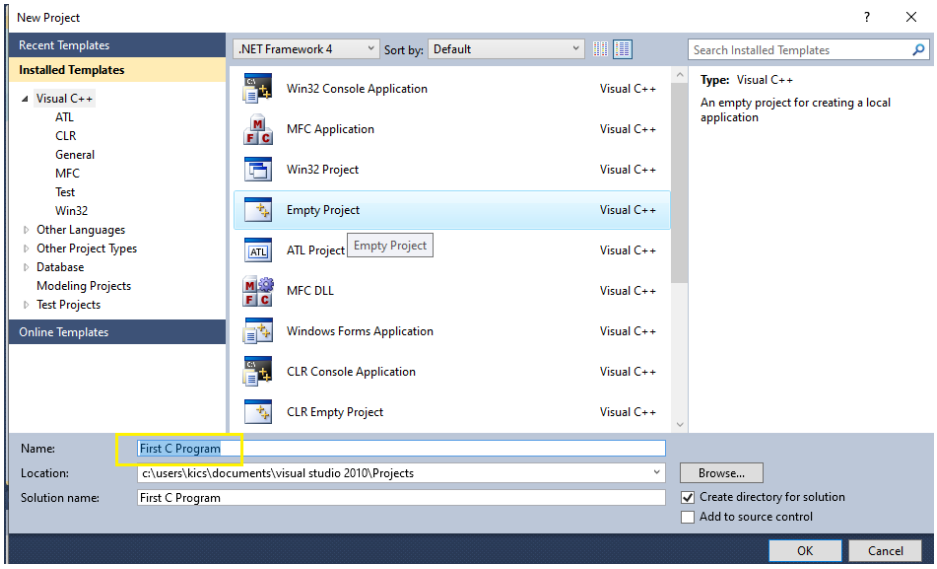


Fig. 5.1 Creating a new project in visual studio

6. Click on add new source (as shown in Figure 5.2).

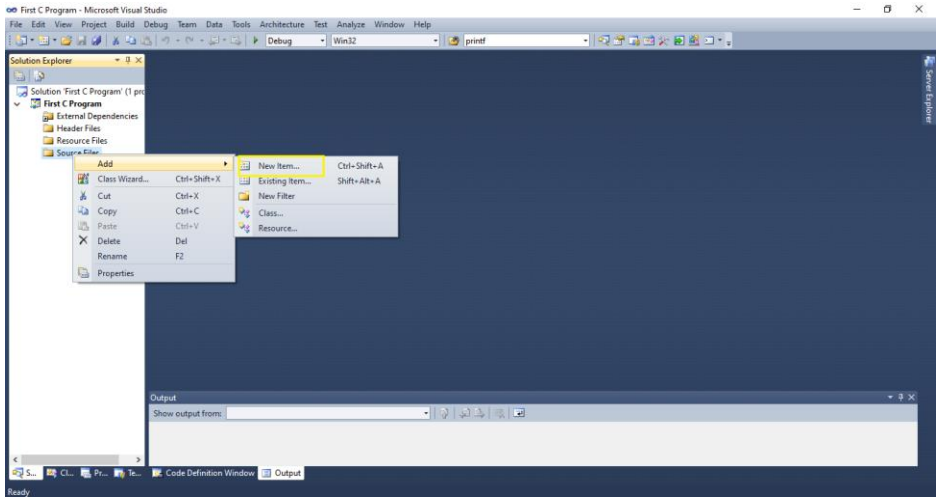


Fig. 5.2 Creating a new Resource File in Visual Studio

7. Select C++ file and name this file (see Figure 5.3).

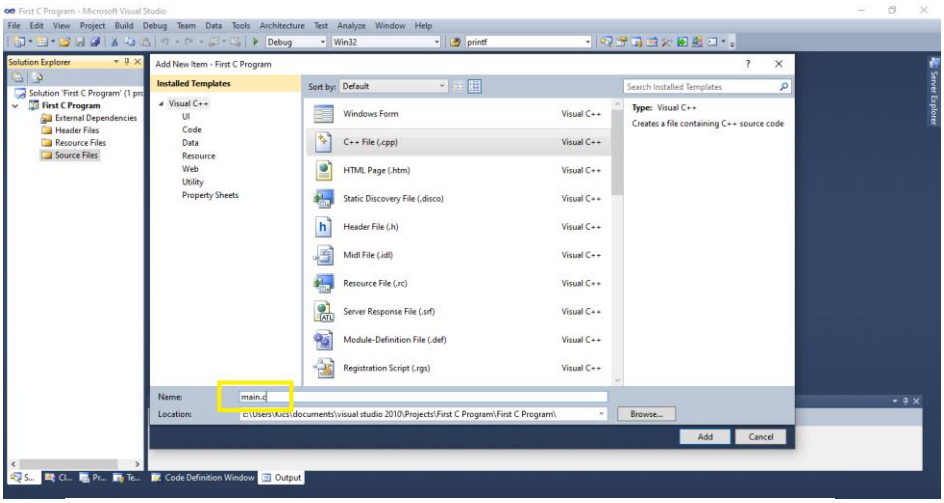


Fig. 5.3 Selecting File Type in Visual Studio

8. Write this code (Figure 5.4) in your source file.

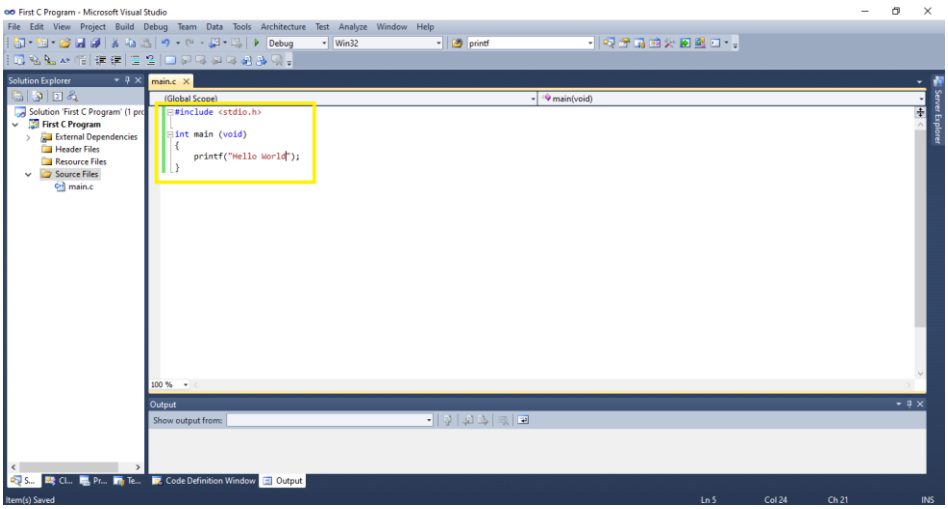


Fig. 5.4 C Program to print Hello World

9. Compile/build the code following the procedure shown in Figure 5.5.

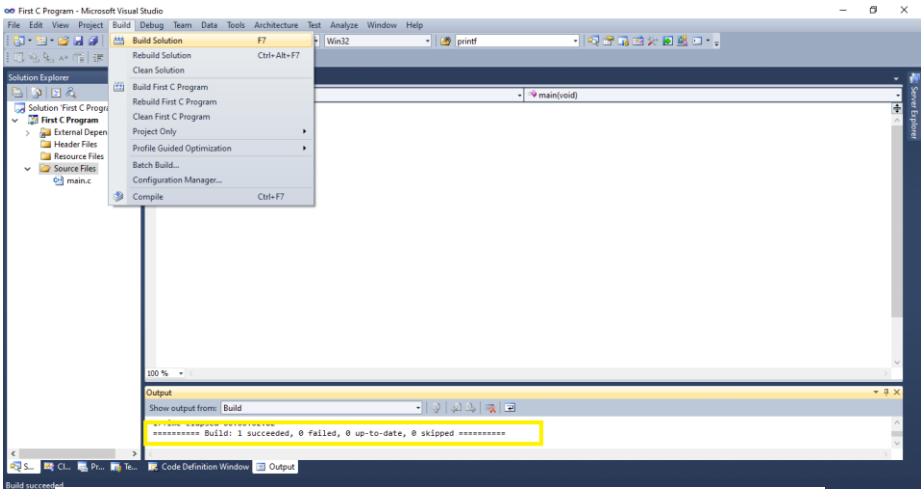


Fig. 5.5 Compiling a C Program in Visual Studio

10. Debug the Program following the procedure in Figure 5.6.

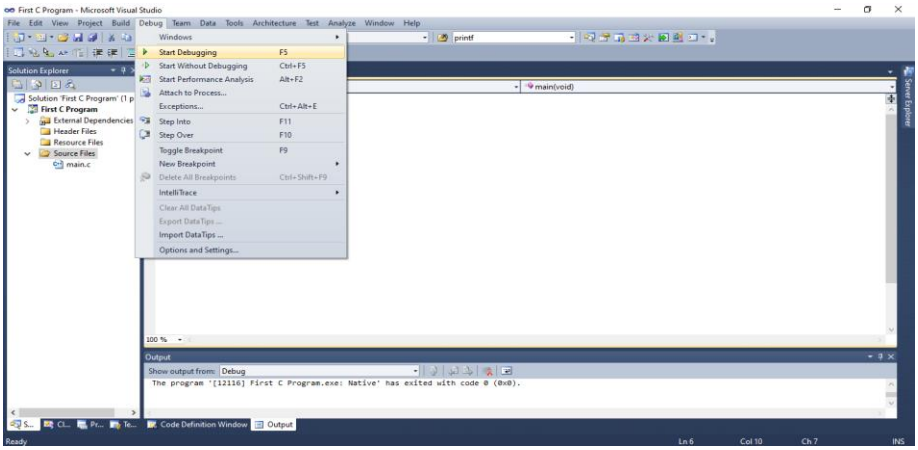


Fig. 5.6 Debugging a C Program in Visual Studio

11. See the output as depicted in Figure 5.7.

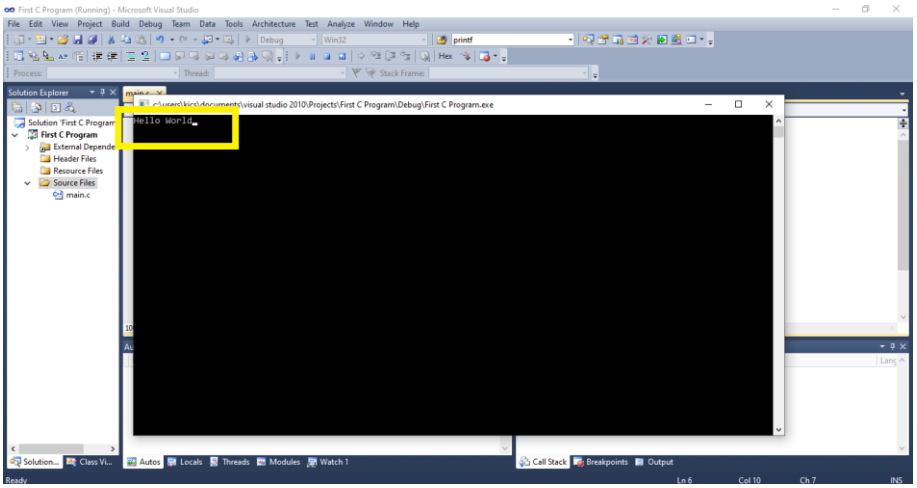


Fig. 5.7 Output of Hello World C Program

Step-in debugging in Microsoft Visual Studio

One very useful feature of Microsoft Visual Studio is the step wise debugging of code. You can run your code line by line to find out the reason for errors rather than running the code at once. Follow Figure 5.8 to perform step-wise debugging.

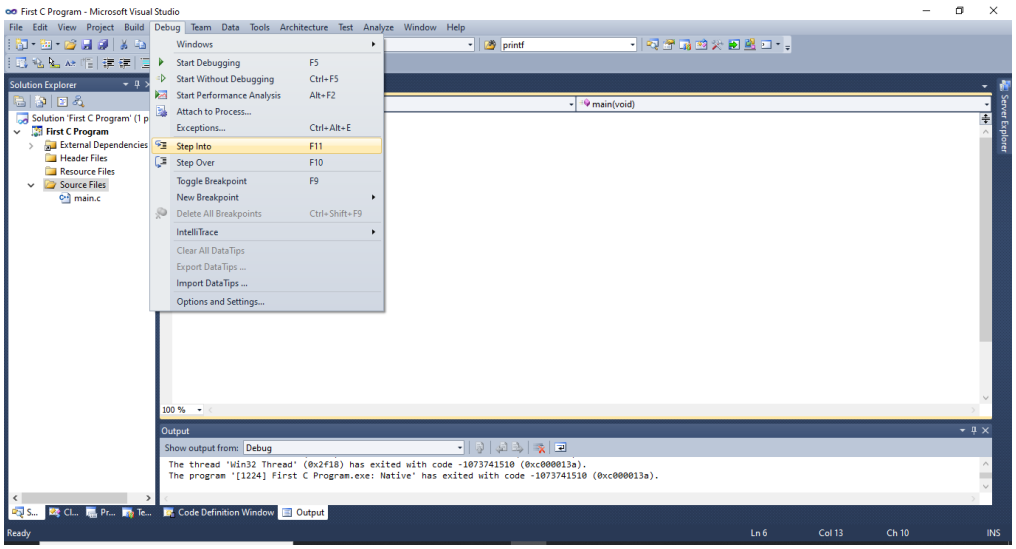


Fig. 5.7 Step-wise Debugging in Visual Studio

5.5 Input/Output Streams in C

Input means feeding some data into a program. An input can be given in the form of a file or from the command line. C programming provides a set of built-in functions to read the given input and feed it to the program as per requirement.

Output means displaying some data on screen, printer, or in any file. C programming provides a set of built-in functions to output the data on the computer screen as well as to save it in text or binary files.

Scanf() and Printf() Functions

The **int scanf(const char *format, ...)** function reads the input from the standard input stream **stdin** and scans that input according to the **format** provided.

The **int printf(const char *format, ...)** function writes the output to the standard output stream **stdout** and produces the output according to the format provided.

The **format** can be a simple constant string, but you can specify %s, %d, %c, %f, etc., to print or read strings, integer, character or float respectively. There are many other formatting options available which can be used based on requirements. We provide a simple example below to understand the concepts better:

```
#include<stdio.h>
int main(){

char str[100];
int i;

   printf("Enter a value :");
   scanf("%s %d", str,&i);

   printf("\nYou entered: %s %d ", str, i);

return0;
}
```

When the above code is compiled and executed, it waits for the user to input some text. When the user enters a text and presses enter, then the program proceeds and reads the input and displays it as follows:

Enter a value: seven 7

You entered: seven 7

Here, it should be noted that scanf() expects input in the same format as you provided %s and %d, which means you have to provide valid inputs like "string integer". If you provide "string string" or "integer integer", then it will be assumed as wrong input.

Secondly, while reading a string, scanf() stops reading as soon as it encounters a space, so "this is test" are three strings for scanf().

Format Specifiers

Format specifiers are used in C for input and output purposes. Using this concept, the compiler can understand that what type of data is in a variable during taking input using the scanf() function and printing using printf() function. Here is a list of format specifiers.

Format Specifier	Type
%c	Character
%d	Signed integer
%f	Float values
%hi	Signed integer (short)
%hu	Unsigned Integer (short)
%i	Unsigned integer
%l or %ld or %li	Long
%lf	Double
%Lf	Long double
%lu	Unsigned int or unsigned long
%lli or %lld	Long long
%llu	Unsigned long long
%o	Octal representation
%p	Pointer
%s	String
%u	Unsigned int

Format Specifier	Type
%x or %X	Hexadecimal representation

These are the basic format specifiers. We can add some other parts with the format specifiers:

- A minus symbol (-) sign tells left alignment
- A number after % specifies the minimum field width. If string is less than the width, it will be filled with spaces
- A period (.) is used to separate field width and precision

Example:

```
#include<stdio.h>
int main(){
   char ch ='B';
   printf("%c\n", ch);//printing character data
   //print decimal or integer data with d and i
   int x =45, y =90;
   printf("%d\n", x);
   printf("%i\n", y);
   float f =12.67;
   printf("%f\n", f);//print float value
   printf("%e\n", f);//print in scientific notation
   int a =67;
   printf("%o\n", a);//print in octal format
   printf("%x\n", a);//print in hex format
   char str[]="Hello World";
   printf("%s\n", str);
   printf("%20s\n", str);//shift to the right 20 characters
including the string
```

```
   printf("%-20s\n", str);//left align

   printf("%20.5s\n", str);//shift to the right 20 characters
including the string, and print string up to 5 character

   printf("%-20.5s\n", str);//left align and print string up to 5
character

}
```

Output

```
B
45
90
12.670000
1.267000e+001
103
43
Hello World
Hello World
Hello World
Hello
Hello
```

We can use these format specifiers for the scanf() function in the same manner.

5.5.1 Data Types

Data types in C refer to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in storage and how the bit pattern stored is interpreted.

Integer Types

The following table provides the details of standard integer types with their storage sizes and value ranges:

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255

signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	8 bytes or (4bytes for 32-bit OS)	-9223372036854775808 to 9223372036854775807
unsigned long	8 bytes	0 to 18446744073709551615

To get the exact size of a type or a variable on a particular platform, you can use the **sizeof** operator. The expressions *sizeof(type)* yields the storage size of the object or type in bytes. Given below is an example to get the size of various type on a machine using different constant defined in limits.h header file:

```
#include<stdio.h>
#include<stdlib.h>
#include<limits.h>
#include<float.h>
int main(int argc,char** argv){
    printf("CHAR_BIT    :   %d\n", CHAR_BIT);
    printf("CHAR_MAX    :   %d\n", CHAR_MAX);
    printf("CHAR_MIN    :   %d\n", CHAR_MIN);
    printf("INT_MAX     :   %d\n", INT_MAX);
    printf("INT_MIN     :   %d\n", INT_MIN);
    printf("LONG_MAX    :   %ld\n",(long) LONG_MAX);
    printf("LONG_MIN    :   %ld\n",(long) LONG_MIN);
    printf("SCHAR_MAX   :   %d\n", SCHAR_MAX);
    printf("SCHAR_MIN   :   %d\n", SCHAR_MIN);
    printf("SHRT_MAX    :   %d\n", SHRT_MAX);
    printf("SHRT_MIN    :   %d\n", SHRT_MIN);
    printf("UCHAR_MAX   :   %d\n", UCHAR_MAX);
    printf("UINT_MAX    :   %u\n",(unsignedint) UINT_MAX);
    printf("ULONG_MAX   :   %lu\n",(unsignedlong) ULONG_MAX);
    printf("USHRT_MAX   :   %d\n",(unsignedshort) USHRT_MAX);
return0;
}
```

When you compile and execute the above program, it produces the following result (specifically for Linux):

```
CHAR_BIT    :   8
CHAR_MAX    :   127
CHAR_MIN    :   -128
INT_MAX     :   2147483647
INT_MIN     :   -2147483648
LONG_MAX    :   9223372036854775807
LONG_MIN    :   -9223372036854775808
SCHAR_MAX   :   127
SCHAR_MIN   :   -128
SHRT_MAX    :   32767
SHRT_MIN    :   -32768
UCHAR_MAX   :   255
UINT_MAX    :   4294967295
ULONG_MAX   :   18446744073709551615
USHRT_MAX   :   65535
```

Floating-Point Types

The following table provides the details of standard floating-point types with storage sizes and value ranges and their precision:

Type	Storage size	Value range	Precision
float	4 byte	1.2E-38 to 3.4E+38	6 decimal places
double	8 byte	2.3E-308 to 1.7E+308	15 decimal places

The header file *float.h* defines macros that allow you to use these values and other details about the binary representation of real numbers in your programs. The following example prints the storage space taken by a float type and its range values:

```c
#include<stdio.h>
#include<stdlib.h>
#include<limits.h>
#include<float.h>

int main(int argc,char** argv){

    printf("Storage size for float : %d \n",sizeof(float));
    printf("FLT_MAX     :   %g\n",(float) FLT_MAX);
    printf("FLT_MIN     :   %g\n",(float) FLT_MIN);
    printf("-FLT_MAX    :   %g\n",(float)-FLT_MAX);
    printf("-FLT_MIN    :   %g\n",(float)-FLT_MIN);
    printf("DBL_MAX     :   %g\n",(double) DBL_MAX);
    printf("DBL_MIN     :   %g\n",(double) DBL_MIN);
    printf("-DBL_MAX     :  %g\n",(double)-DBL_MAX);
    printf("Precision value: %d\n", FLT_DIG );
```

```
return0;
}
```

When you compile and execute the above program, it produces the following result on Linux:

```
Storage size for float : 4
FLT_MAX     :  3.40282e+38
FLT_MIN     :  1.17549e-38
-FLT_MAX    :  -3.40282e+38
-FLT_MIN    :  -1.17549e-38
DBL_MAX     :  1.79769e+308
DBL_MIN     :  2.22507e-308
-DBL_MAX    :  -1.79769e+308
Precision value: 6
```

Void Type

The void type specifies that no value is available. It is used in the following kinds of situations:

Sr.No.	Types & Description
1	**Function returns as void** There are various functions in C which do not return any value or you can say they return void. A function with no return value has the return type as void. For example, **void exit (int status);**
2	**Function arguments as void** There are various functions in C which do not accept any parameter. A function with no parameter can accept a void. For example, **int rand(void);**

5.6 Arithmatic Operators in C

An operator is a symbol that tells the compiler to perform specific mathematical or logical functions. C language is rich in built-in operators and provides the following types of operators −

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Miscellaneous Operators

Arithmetic Operators

The following table shows all the arithmetic operators supported by the C language. Assume variable **A** holds 10 and variable **B** holds 20 then:

Operator	Description	Example
+	Adds two operands.	A + B = 30
−	Subtracts second operand from the first.	A − B = -10
*	Multiplies both operands.	A * B = 200
/	Divides numerator by de-numerator.	B / A = 2
%	Modulus Operator and remainder of after an integer division.	B % A = 0
++	Increment operator increases the integer value by one.	A++ = 11
--	Decrement operator decreases the integer value by one.	A-- = 9

Example

See the following example to understand all the arithmetic operators available in C:

```
#include<stdio.h>

main(){

int a =21;
int b =10;
int c ;

   c = a + b;
   printf("Line 1 - Value of c is %d\n", c );

   c = a - b;
   printf("Line 2 - Value of c is %d\n", c );

   c = a * b;
   printf("Line 3 - Value of c is %d\n", c );

   c = a / b;
   printf("Line 4 - Value of c is %d\n", c );

   c = a % b;
   printf("Line 5 - Value of c is %d\n", c );

   c = a++;
   printf("Line 6 - Value of c is %d\n", c );

   c = a--;
   printf("Line 7 - Value of c is %d\n", c );
}
```

When you compile and execute the above program, it produces the following result −

```
Line 1 - Value of c is 31
Line 2 - Value of c is 11
Line 3 - Value of c is 210
Line 4 - Value of c is 2
Line 5 - Value of c is 1
Line 6 - Value of c is 21
Line 7 - Value of c is 22
```

Relational Operators

The following table shows all the relational operators supported by C. Assume variable **A** holds 10 and variable **B** holds 20 then:

Operator	Description	Example
==	Checks if the values of two operands are equal or not. If yes, then the condition becomes true.	(A == B) is not true.
!=	Checks if the values of two operands are equal or not. If the values are not equal, then the condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand. If yes, then the condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand. If yes, then the condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand. If yes, then the condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand. If yes, then the condition becomes true.	(A <= B) is true.

Example

See the following example to understand all the relational operators available in C:

```
#include<stdio.h>

main(){

int a =21;
int b =10;
int c ;

if( a == b ){
      printf("Line 1 - a is equal to b\n");
}else{
      printf("Line 1 - a is not equal to b\n");
}

if( a < b ){
      printf("Line 2 - a is less than b\n");
}else{
      printf("Line 2 - a is not less than b\n");
}
```

```
if( a > b ){
      printf("Line 3 - a is greater than b\n");
}else{
      printf("Line 3 - a is not greater than b\n");
}

/* Lets change value of a and b */
   a =5;
   b =20;

if( a <= b ){
      printf("Line 4 - a is either less than or equal to  b\n");
}

if( b >= a ){
      printf("Line 5 - b is either greater than  or equal to b\n");
}
}
```

When you compile and execute the above program, it produces the following result:

```
Line 1 - a is not equal to b
Line 2 - a is not less than b
Line 3 - a is greater than b
Line 4 - a is either less than or equal to  b
Line 5 - b is either greater than  or equal to b
```

Logical Operators

The Following table shows all the logical operators supported by C language. Assume variable **A** holds 1 and variable **B** holds 0, then:

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non-zero, then the condition becomes true.	(A && B) is false.
||	Called Logical OR Operator. If any of the two operands is non-zero, then the condition becomes true.	(A || B) is true.
!	Called Logical NOT Operator. It is used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false.	!(A && B) is true.

Example

See the following example to understand all the logical operators available in C:

```
#include<stdio.h>

main(){

int a =5;
int b =20;
int c ;

if( a && b ){
      printf("Line 1 - Condition is true\n");
}

if( a || b ){
      printf("Line 2 - Condition is true\n");
}

/* lets change the value of  a and b */
   a =0;
   b =10;

if( a && b ){
      printf("Line 3 - Condition is true\n");
}else{
      printf("Line 3 - Condition is not true\n");
}

if(!(a && b)){
      printf("Line 4 - Condition is true\n");
}

}
```

When you compile and execute the above program, it produces the following result:

```
Line 1 - Condition is true
Line 2 - Condition is true
Line 3 - Condition is not true
Line 4 - Condition is true
```

Bitwise Operators

Bitwise operator works on bits and perform bit-by-bit operation. The truth tables for &, |, and ^ is as follows:

p	Q	p & q	p | q	p ^ q
0	0	0	0	0
0	1	0	1	1
1	1	1	1	0
1	0	0	1	1

Assume A = 60 and B = 13 in binary format, they will be as follows:
A = 0011 1100
B = 0000 1101

A&B = 0000 1100
A|B = 0011 1101
A^B = 0011 0001
~A = 1100 0011

The following table lists the bitwise operators supported by C. Assume variable 'A' holds 60 and variable 'B' holds 13, then:

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) = 12, i.e., 0000 1100
|	Binary OR Operator copies a bit if it exists in either operand.	(A | B) = 61, i.e., 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) = 49, i.e., 0011 0001
~	Binary One's Complement Operator is unary and has the effect of 'flipping' bits.	(~A) = ~(60), i.e,. -0111101

<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 = 240 i.e., 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 = 15 i.e., 0000 1111

Example

See the following example to understand all the bitwise operators available in C:

```
#include<stdio.h>

main(){

unsignedint a =60;  /* 60 = 0011 1100 */
unsignedint b =13;  /* 13 = 0000 1101 */
int c =0;

   c = a & b;/* 12 = 0000 1100 */
   printf("Line 1 - Value of c is %d\n", c );

   c = a | b;/* 61 = 0011 1101 */
   printf("Line 2 - Value of c is %d\n", c );

   c = a ^ b;/* 49 = 0011 0001 */
   printf("Line 3 - Value of c is %d\n", c );

   c =~a;/*-61 = 1100 0011 */
   printf("Line 4 - Value of c is %d\n", c );

   c = a <<2;/* 240 = 1111 0000 */
   printf("Line 5 - Value of c is %d\n", c );

   c = a >>2;/* 15 = 0000 1111 */
   printf("Line 6 - Value of c is %d\n", c );
}
```

When you compile and execute the above program, it produces the following result:

```
Line 1 - Value of c is 12
Line 2 - Value of c is 61
Line 3 - Value of c is 49
Line 4 - Value of c is -61
Line 5 - Value of c is 240
Line 6 - Value of c is 15
```

Assignment Operators

The following table lists the assignment operators supported by the C language:

Operator	Description	Example
=	Simple assignment operator. Assigns values from right side operands to left side operand	C = A + B will assign the value of A + B to C
+=	Add AND assignment operator. It adds the right operand to the left operand and assign the result to the left operand.	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator. It subtracts the right operand from the left operand and assigns the result to the left operand.	C -= A is equivalent to C = C - A
*=	Multiply AND assignment operator. It multiplies the right operand with the left operand and assigns the result to the left operand.	C *= A is equivalent to C = C * A
/=	Divide AND assignment operator. It divides the left operand with the right operand and assigns the result to the left operand.	C /= A is equivalent to C = C / A
%=	Modulus AND assignment operator. It takes modulus using two operands and assigns the result to the left operand.	C %= A is equivalent to C = C % A
<<=	Left shift AND assignment operator.	C <<= 2 is same as C = C << 2
>>=	Right shift AND assignment operator.	C >>= 2 is same as C = C >> 2
&=	Bitwise AND assignment operator.	C &= 2 is same as C = C & 2
^=	Bitwise exclusive OR and assignment operator.	C ^= 2 is same as C = C ^ 2
|=	Bitwise inclusive OR and assignment operator.	C |= 2 is same as C = C | 2

Example

Try the following example to understand all the assignment operators available in C:

```
#include<stdio.h>

main(){

int a =21;
int c ;

   c =  a;
   printf("Line 1 - =  Operator Example, Value of c = %d\n", c );

   c +=  a;
   printf("Line 2 - += Operator Example, Value of c = %d\n", c );

   c -=  a;
   printf("Line 3 - -= Operator Example, Value of c = %d\n", c );

   c *=  a;
   printf("Line 4 - *= Operator Example, Value of c = %d\n", c );

   c /=  a;
   printf("Line 5 - /= Operator Example, Value of c = %d\n", c );

   c  =200;
   c %=  a;
   printf("Line 6 - %= Operator Example, Value of c = %d\n", c );

   c <<=2;
   printf("Line 7 - <<= Operator Example, Value of c = %d\n", c );

   c >>=2;
   printf("Line 8 - >>= Operator Example, Value of c = %d\n", c );

   c &=2;
   printf("Line 9 - &= Operator Example, Value of c = %d\n", c );

   c ^=2;
   printf("Line 10 - ^= Operator Example, Value of c = %d\n", c );

   c |=2;
   printf("Line 11 - |= Operator Example, Value of c = %d\n", c );
}
```

When you compile and execute the above program, it produces the following result:

```
Line 1 - =  Operator Example, Value of c = 21
```

```
Line 2 - += Operator Example, Value of c = 42
Line 3 - -= Operator Example, Value of c = 21
Line 4 - *= Operator Example, Value of c = 441
Line 5 - /= Operator Example, Value of c = 21
Line 6 - %= Operator Example, Value of c = 11
Line 7 - <<= Operator Example, Value of c = 44
Line 8 - >>= Operator Example, Value of c = 11
Line 9 - &= Operator Example, Value of c = 2
Line 10 - ^= Operator Example, Value of c = 0
Line 11 - |= Operator Example, Value of c = 2
```

Operators Precedence in C

Operator precedence determines the grouping of terms in an expression and decides how an expression is evaluated. Certain operators have higher precedence than others; for example, the multiplication operator has a higher precedence than the addition operator.

For example, x = 7 + 3 * 2; here, x is assigned 13, not 20 because operator * has a higher precedence than +, so it first gets multiplied with 3*2 and then adds into 7.

Here, operators with the highest precedence appear at the top of the table, those with the lowest appear at the bottom. Within an expression, higher precedence operators will be evaluated first.

Category	Operator	Associativity
Postfix	() [] -> . ++ - -	Left to right
Unary	+ - ! ~ ++ - - (type)* & sizeof	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Shift	<<>>	Left to right
Relational	<<= >>=	Left to right
Equality	== !=	Left to right
Bitwise AND	&	Left to right
Bitwise XOR	^	Left to right
Bitwise OR	|	Left to right

Logical AND	&&	Left to right
Logical OR	||	Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %=>>= <<= &= ^= |=	Right to left
Comma	,	Left to right

Example

Understand the following example to understand operator precedence in C:

```
#include<stdio.h>

main(){

int a =20;
int b =10;
int c =15;
int d =5;
int e;

   e =(a + b)* c / d;// ( 30 * 15 ) / 5
   printf("Value of (a + b) * c / d is : %d\n",  e );

   e =((a + b)* c)/ d;// (30 * 15 ) / 5
   printf("Value of ((a + b) * c) / d is  : %d\n",  e );

   e =(a + b)*(c / d);// (30) * (15/5)
   printf("Value of (a + b) * (c / d) is  : %d\n",  e );

   e = a +(b * c)/ d;//  20 + (150/5)
   printf("Value of a + (b * c) / d is  : %d\n",  e );

return0;
}
```

When you compile and execute the above program, it produces the following result:

```
Value of (a + b) * c / d is : 90
Value of ((a + b) * c) / d is  : 90
Value of (a + b) * (c / d) is  : 90
Value of a + (b * c) / d is  : 50
```

5.7 Decision Making

Decision making structures require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed. If the condition is determined to be true, some specific statements are executed and if false, then other set of statements are executed.

Shown below is the general form of a typical decision-making structure found in most of the programming languages:

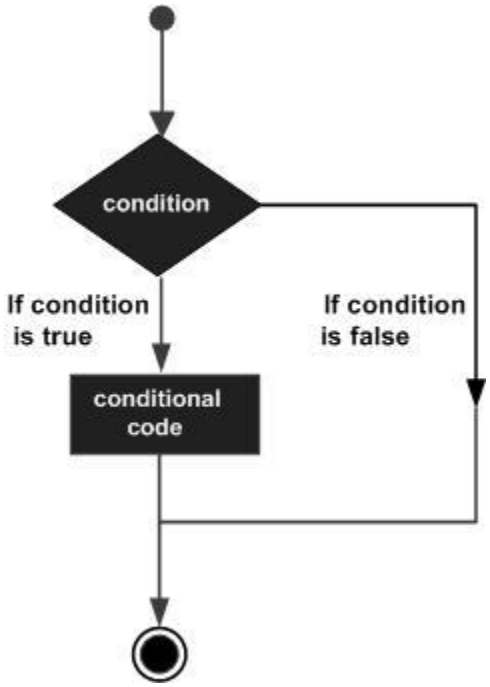


Fig. 5.8 Flowchart of Decision Making in C

C programming language assumes any **non-zero** and **non-null** values as **true**, and if it is either **zero** or **null**, then it is assumed as **false** value. C programming language provides the following types of decision-making statements:

Sr.No.	Statement & Description
1	if statement An **if statement** consists of a boolean expression followed by one or more statements.
2	if...else statement An **if statement** can be followed by an optional **else statement**, which executes when the Boolean expression is false.

3	nested if statements You can use one **if** or **else if** statement inside another **if** or **else if** statement(s).
4	switch statement A **switch** statement allows a variable to be tested for equality against a list of values.
5	nested switch statements You can use one **switch** statement inside another **switch** statement(s).

If-else Statement

An **if** statement can be followed by an optional **else** statement, which executes when the Boolean expression is false.

Syntax

The syntax of an **if...else** statement in C programming language is:

```
if(boolean_expression) {
   /* statement(s) will execute if the boolean expression is true
*/
} else {
   /* statement(s) will execute if the boolean expression is false
*/
}
```

If the Boolean expression evaluates to **true**, then the **if block** will be executed, otherwise, the **else block** will be executed.

C programming language assumes any **non-zero** and **non-null** values as **true**, and if it is either **zero** or **null**, then it is assumed as **false** value.

Example

```
#include<stdio.h>

int main (){

/* local variable definition */
int a =100;

/* check the boolean condition */
if( a <20){
/* if condition is true then print the following */
```

```
      printf("a is less than 20\n");
}else{
/* if condition is false then print the following */
      printf("a is not less than 20\n");
}

   printf("value of a is : %d\n", a);

return0;
}
```

When the above code is compiled and executed, it produces the following result:

```
a is not less than 20;
value of a is : 100
```

Nested If Statements

It is always legal in C programming to **nest** if-else statements, which means you can use one if or else if statement inside another if or else if statement(s).

Syntax

The syntax for a **nested if** statement is as follows:

```
if( boolean_expression 1) {

   /* Executes when the boolean expression 1 is true */
   if(boolean_expression 2) {
      /* Executes when the boolean expression 2 is true */
   }
}
```

You can nested **else if...else** in the similar way as you have nested *if* statements.

Example

```
#include<stdio.h>

int main (){

/* local variable definition */
int a =100;
int b =200;

/* check the boolean condition */
if( a ==100){
```

```
/* if condition is true then check the following */
if( b ==200){
/* if condition is true then print the following */
         printf("Value of a is 100 and b is 200\n");
}
}

   printf("Exact value of a is : %d\n", a );
   printf("Exact value of b is : %d\n", b );

return0;
}
```

When the above code is compiled and executed, it produces the following result:

```
Value of a is 100 and b is 200
Exact value of a is : 100
Exact value of b is : 200
```

Switch Statements

A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each **switch case**.

Syntax

The syntax for a **switch** statement in C programming language is as follows:

```
switch(expression){

case constant-expression  :
      statement(s);
break;/* optional */

case constant-expression  :
      statement(s);
break;/* optional */

/* you can have any number of case statements */
default:/* Optional */
   statement(s);
}
```

The following rules apply to a **switch** statement:

- The **expression** used in a **switch** statement must have an integral or enumerated type, or be of a class type in which the class has a single conversion function to an integral or enumerated type.
- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
- The **constant-expression** for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.
- When the variable being switched on is equal to a case, the statements following that case will execute until a **break** statement is reached.
- When a **break** statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
- Not every case needs to contain a **break**. If no **break** appears, the flow of control will *fall through* to subsequent cases until a break is reached.
- A **switch** statement can have an optional **default** case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No **break** is needed in the default case.

Example

```
#include<stdio.h>

int main (){

/* local variable definition */
char grade ='B';

switch(grade){
case'A':
        printf("Excellent!\n");
break;
case'B':
case'C':
        printf("Well done\n");
```

```
break;
case'D':
         printf("You passed\n");
break;
case'F':
         printf("Better try again\n");
break;
default:
         printf("Invalid grade\n");
}

   printf("Your grade is  %c\n", grade );

return0;
}
```

When the above code is compiled and executed, it produces the following result:

```
Well done
Your grade is B
```

5.8 Loops in C

Loops are used when a block of code needs to be executed several times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on. Programming languages provide various control structures that allow for more complicated execution paths.

C programming language provides the following types of loops to handle looping requirements.

Loop Type & Description
while loop Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
for loop Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
do...while loop It is more like a while statement, except that it tests the condition at the end of the loop body.
nested loops You can use one or more loops inside any other while, for, or do..while loop.

While Loop

A **while** loop in C programming repeatedly executes a target statement as long as a given condition is true.

Syntax

The syntax of a **while** loop in C programming language is:

```
while(condition) {

   statement(s); }
```

Here, **statement(s)** may be a single statement or a block of statements. The **condition** may be any expression, and true is any nonzero value. The loop iterates while the condition is true.

When the condition becomes false, the program control passes to the line immediately following the loop.

Example

```
#include<stdio.h>

int main (){

/* local variable definition */
int a =10;

/* while loop execution */
while( a <20){
      printf("value of a: %d\n", a);
      a++;
}

return0;
}
```

When the above code is compiled and executed, it produces the following result:

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

For Loop

A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Syntax

The syntax of a **for** loop in C programming language is:

```
for ( init; condition; increment ) {
   statement(s);
}
```

Below is the flow of control in a 'for' loop −

- The **init** step is executed first, and only once. This step allows you to declare and initialize any loop control variables. You are not required to put a statement here, as long as a semicolon appears.
- Next, the **condition** is evaluated. If it is true, the body of the loop is executed. If it is false, the body of the loop does not execute and the flow of control jumps to the next statement just after the 'for' loop.
- After the body of the 'for' loop executes, the flow of control jumps back up to the **increment** statement. This statement allows you to update any loop control variables. This statement can be left blank, as long as a semicolon appears after the condition.
- The condition is now evaluated again. If it is true, the loop executes and the process repeats itself (body of loop, then increment step, and then again condition). After the condition becomes false, the 'for' loop terminates.

Example

```
#include<stdio.h>

int main (){

int a;

/* for loop execution */
for( a =10; a <20; a = a +1){
```

```
      printf("value of a: %d\n", a);
}

return0;
}
```

When the above code is compiled and executed, it produces the following result:

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

Loop Control Statements

Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed. C supports the following control statements.

Control Statement & Description
break statement Terminates the **loop** or **switch** statement and transfers execution to the statement immediately following the loop or switch.
continue statement Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.
goto statement Transfers control to the labeled statement.

The **break** statement in C programming has the following two usages −

- When a **break** statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.
- It can be used to terminate a case in the **switch** statement (covered in the next chapter).

If you are using nested loops, the break statement will stop the execution of the innermost loop and start executing the next line of code after the block.

Syntax

The syntax for a **break** statement in C is as follows:

break;

5.9 Functions in C

A function is a group of statements that together perform a task. Every C program has at least one function, which is **main()**, and all the most trivial programs can define additional functions. You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division is such that each function performs a specific task.

A function **declaration** tells the compiler about a function's name, return type, and parameters. A function **definition** provides the actual body of the function.

The C standard library provides numerous built-in functions that your program can call. For example, **strcat()** to concatenate two strings, **memcpy()** to copy one memory location to another location, and many more functions.

A function can also be referred as a method or a sub-routine or a procedure, etc.

Defining a Function

The general form of a function definition in C programming language is as follows:

```
return_type function_name( parameter list ){
   body of the function
}
```

The following are the parts of a function −

- **Return Type** − A function may return a value. The **return_type** is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return_type is the keyword **void**.
- **Function Name** − This is the actual name of the function. The function name and the parameter list together constitute the function signature.

- **Parameters** − A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.
- **Function Body** − The function body contains a collection of statements that define what the function does.

Example

Given below is the source code for a function called **max()**. This function takes two parameters num1 and num2 and returns the maximum value between the two.

```
/* function returning the max between two numbers */
int max(int num1,int num2){

/* local variable declaration */
int result;

if(num1 > num2)
      result = num1;
else
      result = num2;

return result;
}
```

Function Declarations

A function **declaration** tells the compiler about a function name and how to call the function. The actual body of the function can be defined separately. A function declaration has the following parts:

return_type function_name(parameter list);

For the above defined function max(), the function declaration is as follows −

int max(int num1, int num2);

Parameter names are not important in function declaration only their type is required, so the following is also a valid declaration:

int max(int, int);

Function declaration is required when you define a function in one source file and you call that function in another file. In such case, you should declare the function at the top of the file calling the function.

Calling a Function

To use a function, you will have to call that function to perform the defined task. When a program calls a function, the program control is transferred to the called function. A called function performs a defined task and when its return statement is executed or when its function-ending closing brace is reached, it returns the program control back to the main program.

To call a function, you simply need to pass the required parameters along with the function name, and if the function returns a value, then you can store the returned value. For example:

```
#include<stdio.h>

/* function declaration */
int max(int num1,int num2);

int main (){

/* local variable definition */
int a =100;
int b =200;
int ret;

/* calling a function to get max value */
   ret = max(a, b);

   printf("Max value is : %d\n", ret );

return0;
}

/* function returning the max between two numbers */
int max(int num1,int num2){

/* local variable declaration */
int result;

if(num1 > num2)
      result = num1;
```

```
else
      result = num2;

return result;
}
```

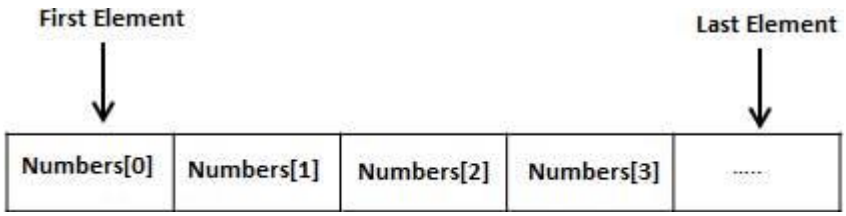
We have kept max() along with main() and compiled the source code. While running the final executable, it would produce the following result:

```
Max value is : 200
```

5.10 Arrays in C

Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type. Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index.

All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.



Declaring Arrays

To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows:

```
type arrayName [ arraySize ];
```

This is called a *single-dimensional* array. The **arraySize** must be an integer constant greater than zero and **type** can be any valid C data type. For example, to declare a 10-element array called **balance** of type double, use this statement:

```
double balance[10];
```

Here *balance* is a variable array which is sufficient to hold up to 10 double numbers.

Initializing Arrays

You can initialize an array in C either one by one or using a single statement as follows:

```
double balance[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

The number of values between braces { } cannot be larger than the number of elements that we declare for the array between square brackets [ ].

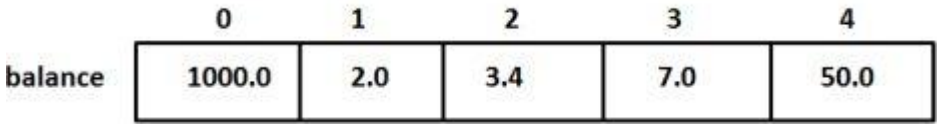
If you omit the size of the array, an array just big enough to hold the initialization is created. Therefore, if you write:

```
double balance[] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

You will create exactly the same array as you did in the previous example. Following is an example to assign a single element of the array:

```
balance[4] = 50.0;
```

The above statement assigns the 5th element in the array with a value of 50.0. All arrays have 0 as the index of their first element which is also called the base index and the last index of an array will be total size of the array minus 1. Shown below is the pictorial representation of the array we discussed above:



Accessing Array Elements

An element is accessed by indexing the array name. This is done by placing the index of the element within square brackets after the name of the array. For example:

double salary = balance[9];

The above statement will take the 10th element from the array and assign the value to salary variable. The following example Shows how to use all the three above mentioned concepts viz. declaration, assignment, and accessing arrays:

```
#include<stdio.h>

int main (){

int n[10];/* n is an array of 10 integers */
int i,j;
```

```
/* initialize elements of array n to 0 */
for( i =0; i <10; i++){
      n[ i ]= i +100;/* set element at location i to i + 100 */
}

/* output each array element's value */
for(j =0; j <10; j++){
      printf("Element[%d] = %d\n", j, n[j]);
}

return0;
}
```

When the above code is compiled and executed, it produces the following result:

```
Element[0] = 100
Element[1] = 101
Element[2] = 102
Element[3] = 103
Element[4] = 104
Element[5] = 105
Element[6] = 106
Element[7] = 107
Element[8] = 108
Element[9] = 109
```

5.11 Strings in C

Strings are actually one-dimensional array of characters terminated by a **null** character '\0'. Thus, a null-terminated string contains the characters that comprise the string followed by a **null**.

The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello."

```
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

If you follow the rule of array initialization then you can write the above statement as follows:

```
char greeting[] = "Hello";
```

There is no need to place the *null* character at the end of a string constant. The C compiler automatically places the '\0' at the end of the string when it initializes the array. Let us try to print the above mentioned string:

```
#include<stdio.h>

int main (){

char greeting[6]={'H','e','l','l','o','\0'};
   printf("Greeting message: %s\n", greeting );
return0;
}
```

When the above code is compiled and executed, it produces the following result:

```
Greeting message: Hello
```

Teacher Notes:

- Inbuilt string handling functions can also be taught i.e., strcpy and strtok etc.

Exercise

Select the most appropriate option.

1. Which of the following characters are used to comment in C.
   a. \\b. /* */ c. % d. Both a and b

2. Which one of the following is a format specifier of *char*.

a. %db.&l c. %c d. Both a and b

3. Which of the following functions is used to take inputs from users.
   a. sprintf() b. printf() c. scanf() d. None of these

4. Which one of the following is a format specifier of *string*.

a. %s b. &l c. %c d. Both a and b

5. Which one of the following is a format specifier of *int*.

a. %d b. &l c. %c d. Both a and b

Answer the following question.

1. Write a program to take input of name, rollno and marks obtained by a student in 5 subjects.Each subject have 100 full marks.Display the name, roll no with percentage score secured.
2. Write a program to add two numbers (5&7) and display its sum.
3. Write a program to input marks of 5 subjects (Physics, Chemistry, Math, English & Biology) for a student. Display the rank of each subjects and also the result of total marks and percentage obtained with his/her rank in the class. The rank is categorized as fail (marks < 40%), pass & third division (marks between 40 to

55%), second (marks between 55 to 65%), first (marks between 65 to 80%), Distinction (marks between 80 to 95%), extra ordinary (marks above 95 to 100%).

4. Write a program to input two integer numbers and display the sum of even numbers between these two input numbers.
5. Write a program to add, subtract, multiply and divide two integers using user defined type function with return type.
6. Write a program to find separately the sum of the positive and negative integer elements of an array of size 10. Define a function called sortarray(int[]), pass the array to this function and display the array elements into ascending order.
7. Write a program to read a sentence and count the number of characters &words in that sentence.

Chapter 6
Computer Security

After Studying this chapter, you will be able to

- differentiate computer security and cyber security.
- understand information system and information system assets.
- understand threats to information system assets.
- describe attack vector, understand malware and its type.
- understand the basic of: confidentiality, integrity, authentication, availability, and accountability.
- understand basic techniques used for confidentiality, integrity and availability.
- understand protection and various tools to protect information system from intruders and hacker.
- download and install anti malware software online and from DVD/CD.
- update anti malware database/repository and perform complete malware scan on any system.
- delete / quarantine all the malware successfully which are detected as a result of scan.
- know well-known firewalls and knowledge of: users account, privileges setting, IP, zones etc.
- describe process of configuring a firewall for: deletion of default account, renaming an account.
- ensuring password protection and applying privileges setting.
- setting inbound and outbound setting and its importance, access control list.
- knowledge of logging services in firewall, reports and logs and explain its importance.
- know about encryption and decryption, encryption tools used in windows.
- perform encryption of drive data using encryption tool.
- understand backup and restore and know importance of schedule backup.
- describe tools used in windows for backup and restore.
- perform backup a data and perform restoring a data.

6.1 Computer Security and Cyber Security

Computer security means dealing with the security of a computer's software and hardware. One of the most important aspects of computer security is keeping stand-alone machines updated and properly patched.

Cyber security, on the other hand, involves the security of individual computers plus the network they belong to and the intercommunication between all the computers in an organization. It can be easy to remember the difference by knowing that the meaning of the word 'cyber' is a 'culture of computers.' It encompasses everything that is connected.

Common Security Threats

There is always an ever-evolving list of security threats and dangers. Some of the most common cyber threats are:

Virusesare malicious programs that get loaded onto victims' computers without their knowledge or permission. Typically, they replicate and infect files and programs installed on the computer.

Worms are software programs that will copy themselves from one computer to another, automatically. They use up disk space and can quickly replicate with larger volumes and speed.

Keylogger or a keystroke logger tracks the real-time activity of someone using their computer. It records every keystroke made on a keyboard and is a powerful threat as it can steal individuals' usernames, passwords and login credentials.

Botnet are when a group of computers connected online have all been compromised by a hacker using a virus. The victims' computers are typically then used for malicious activities and bigger attacks such as denial-of-service attacks.

Phishing is when criminal disguises themselves as a business or individual to steal sensitive data or personal information, typically via fraudulent emails and malicious links. Phishing is so dangerous because it is easy to execute and hackers are getting more sophisticated at making their emails look legitimate.

6.2 Information Systems

Information systems incorporate computers to help manage data and achieve business objectives, they do not necessarily have to include computers. There are different types of information systems that can serve a variety of purposes depending on an organization's needs. Examples include:

- **Data warehouses:** Data warehouses are data management systems that support analytics and other business intelligence activities. They consolidate and analyze data from a large variety of sources. Data warehouses can provide insight into businesses to help improve decision-making.
- **Enterprise systems:** Enterprise systems, also known as enterprise resource planning (ERP) systems, are integrated systems that combine all the hardware and software a business uses for different functions in its operations. These organization-wide systems help information flow between departments and allow processes from different parts of the business to be integrated across a company.
- **Expert systems:** Expert systems use artificial intelligence to mimic human decision-making. The software uses human knowledge to solve problems that would typically require a person's expertise. Expert systems can be applied in areas such as medical diagnoses, accounting and coding.
- **Geographic information systems:** Geographic information systems (GIS) are tools that gather, organize, map and analyze data with a spatial component. GIS can improve analysis and decision-making by allowing users to visualize data on a map. Global information systems are a type of GIS that synthesize worldwide data.
- **Office automation systems:** Office automation systems combine communication technology, people and computers to help perform office activities, such as preparing written communication, printing, scheduling or creating reports.

6.2.1 Components of Information Systems

Every information system includes several key components: hardware, software, telecommunications, people and data.

- **Hardware** refers to the physical pieces of the information system
- **Software** is the programming that controls the information system
- **Telecommunication** transmits information through the system
- **Humans** manage and interact with the information system
- **Data** is information stored within and processed by the system.

Hardware

The hardware component of an information system comprises the physical elements of the system. People can touch and feel pieces of hardware. These mechanisms, equipment and wiring allow systems like computers, smartphones and tablets to function.

Input and output devices are essential pieces of technology that allow humans to interact with computers and other information systems. Keyboards, mice, microphones and scanners are all examples of input devices.

Output devices might include printers, monitors, speakers and sound and video cards. Pieces of hardware including microprocessors, hard drives, electric power supply units, and removable storage also allow computers to store and process data.

Software

Software are the intangible programs that manage information system functions, including input, output, processing and storage. System software – such as the MacOS or Microsoft Windows operating systems – provides a base for application software to run.Application software operates programs for particular uses in information systems. For instance, word processing applications are used to create and edit text documents. Graphical user interface (GUI) software is among the most common application software.It presents the information stored in computers and allows users to interact with computers through digital graphics – such as icons, buttons and scroll bars – rather than through text-based commands.

Software can be either open source or closed source. Open-source software coding is publicly available for users and programmers to manipulate, whereas closed source software is proprietary.

Telecommunications

Telecommunications systems connect computer networks and allow information to be transmitted through them. Telecommunications networks also allow computers and storage services to access information from the cloud.

There are a number of methods telecommunications networks use to convey information. Coaxial cables and fiber optic cables are used by telephone, internet and cable providers to transmit data, video and audio messages.

Local-area networks (LANs) connect computers to create computer networks in a designated space, like a school or home. Wide-area networks (WANs) are collections of LANs that facilitate data-sharing across large areas. A virtual private network (VPN) allows a user to protect their online privacy by encrypting data on public networks. Microwaves and radio waves can also be used to transmit information in telecommunications networks.

Data

Data are intangible, raw facts that are stored, transmitted, analysed and processed by other components of information systems. Data are often stored as numerical facts, and they represent quantitative or qualitative information. Data can be stored in a database or data warehouse, in a form that best suits the organization using it.

Databases contain collections of data that can be queried or retrieved for specific purposes. Databases allow users to perform fundamental operations, such as storage and retrieval. Data warehouses, on the other hand, store data from multiple sources for analytical purposes. They allow users to assess an organization or its operations.

Human Resources

Human resources are a crucial part of information systems. The human component of information systems encompasses the qualified people who influence and manipulate the data in information systems. Humans involved in information systems may

include business analysts, information security analysts or system analysts.Business analysts work to elevate an organization's operations and processes. They often focus on improving efficiency and productivity or streamlining distribution. Information security analysts work to prevent data breaches and cybersecurity attacks.

6.3 Importance of Computer Security

Computer Security is, basically the preservation of computing systems and the information that they save and / or retrieve. Computer Security is important as it enables people to perform their work and study. It helps in the execution of essential business processes, and safeguards confidential and sensitive information.

Attack Vector

An attack vector is a path or means by which an attacker or hacker can gain access to a computer or network server in order to deliver a payload or malicious outcome. Attack vectors enable hackers to exploit system vulnerabilities, including the human element.Common cyber attack vectors include viruses and malware (intrusive software), email attachments, webpages, pop-up windows, instant messages (IMs), chatrooms and deception. Deception is when a human operator is fooled into removing or weakening system defences.Firewalls and antivirus software can block attack vectors. But no protection method is totally attack-proof.

6.4 Losses in case of Security Attacks

Productivity Loss

Losses that result from an organization's inability to deliver its products or services.

Response Loss

Losses that are associated with managing the event itself. This form of loss will be the most common across your analyses.

Replacement Loss

The costs associated with the replacement of a capital asset or a person.

Fines and Judgments

Penalties levied against an organization through civil, criminal or contractual actions, usually the result of a Confidentiality related scenario.

Reputation Damage

Losses associated with an external actor's perception of your organization whereby its main value proposition is diminished.

6.5 Difference between hacker and a Cracker

There are a lot of dissimilarities between hackers and crackers, and one should not use them interchangeably. The main difference between hackers and crackers can be observed through the following points:

- Hackers are people who use their knowledge for a good purpose and do not damage the data, whereas a cracker is someone who breaks into the system with a malicious purpose and damages data intentionally.
- Hackers possess advanced knowledge of computer systems and programming languages, while crackers might not necessarily be so skilled and well-versed with computing knowledge.
- The hackers work for an organization to improvise their network and solve any issues. Crackers are someone from whom the hacker protects the organization. Crackers work just because a system might be challenging or to get illegal gains.
- Hacking is ethical, while cracking is illegal and unethical.
- Hackers have ethical certificates, while the Crackers do not possess any certificates.
- Hackers continuously work towards making new tools rather than using the existing ones. The crackers, on the other hand, have inadequate computing knowledge to make new tools and use tools already used by other crackers.

Do you know

- In common language, the word *hacker* is mostly used instead of *cracker*. In this book, the word *hacker* is used throughout instead of *cracker*. While, Section 6.5 describes their differences.

Risk vs. threat vs. vulnerability

Risk is any possibility for loss, damage or destruction of assets or data caused by a cyber threat. ***Threat*** is a process that magnifies the likelihood of a negative event, such as the exploit of a vulnerability. A ***vulnerability*** is a weakness in your infrastructure, networks or applications that potentially exposes you to threats. When a threat targets a vulnerability that exists in your IT infrastructure, network or applications, it can result in risk to your assets, data or business.

Risk: An organization's risk profile fluctuates depending on internal and external environmental factors. It incorporates not just the potential or probability of a negative event, but the impact that event may have on your infrastructure. And though risk can never be 100% eliminated—cybersecurity is a persistently moving target, after all—it can be managed to a level that satisfies your organization's tolerance for risk.

Threat: Today's cybersecurity landscape roils with an endless stream of potential threats—from malware that plants dangerous executables in your software and ransomware that locks up your systems to specially targeted hacker attacks. All of these threats look for a way in, a vulnerability in your environment that they can exploit. Some threats, however, hold more potential for exploitation than others.

Vulnerability: Vulnerabilities are weak spots within your environment and your assets—weaknesses that open you up to potential threats and increased risk. And unfortunately, an organization can have thousands, often millions of vulnerabilities.

6.6 Features of a Secure System

Cybersecurity is the responsibility of every employee who has or accesses a computer that belongs to the company. Standardized rules are essential so that everyone knows what is expected of them. Here are some critical features of a sound cybersecurity policy:

Confidentiality

Confidentiality means protecting the information from being exposed to an unauthorized party due to a data breach or insider threat. Nobody wants to deal with the fallout of a data breach, which is why you should take major steps to implement document security,

establish security controls for sensitive files, and establish clear information security policies regarding devices. Confidentiality covers a spectrum of access controls and measures that protect your information from getting misused by any unauthorized access. The ideal way to keep your data confidential and prevent a data breach is to implement safeguards.

A failure to maintain confidentiality means that someone who shouldn't have access has managed to get access to private information. Through intentional behaviour or by accident. A failure in confidentiality can cause some serious devastation.

Some information security basics to keep your data confidential are:

1. Encryption
2. Password
3. Two-factor authentication
4. Biometric verification

Integrity

Integrity refers to the accuracy and completeness of data. Security controls focused on integrity are designed to prevent data from being modified or misused by an unauthorized party. Integrity involves maintaining the consistency and trustworthiness of data over its entire life cycle. Data must not be changed in transit, and precautionary steps must be taken to ensure that data cannot be altered by unauthorized people. For example, in a data breach that compromises integrity, a hacker may seize data and modify it before sending it on to the intended recipient. Some security controls designed to maintain the integrity of information include:

1. Encryption
2. User access controls
3. Version control
4. Backup and recovery procedures
5. Error detection software

Availability

Data availability means that information is accessible to authorized users. It provides an assurance that your system and data can be accessed by authenticated users whenever they're needed. Similar to confidentiality and integrity, availability also holds great value. Availability is typically associated with reliability and system uptime, which can be impacted by non-malicious issues like hardware failures, unscheduled software downtime, and human error, or malicious issues like cyberattacks and insider threats. If the network goes down unexpectedly, users will not be able to access essential data and applications.

Making regular off-site backups can limit the damage caused to hard drives by natural disasters or server failure. Information only has value if the right people can access it at the right time. Information security measures for mitigating threats to data availability include:

1. Off-site backups
2. Disaster recovery
3. Redundancy
4. Failover
5. Proper monitoring
6. Environmental controls
7. Virtualization
8. Server clustering

Authentication

Authentication is the process of identifying users that request access to a system, network, or device. Access control determines user identity according to credentials like username and password. Other authentication technologies like biometrics and authentication apps are also used to authenticate user identity.

Non-repudiation

Non-repudiation is the assurance that someone cannot deny the validity of something. Non-repudiation is a legal concept that is widely used in information security and refers to a service, which provides proof of the origin of data and the integrity of the data.

Digital signatures (combined with other measures) can offer non-repudiation when it comes to online transactions, where it is crucial to ensure that a party to a contract or a communication can't deny the authenticity of their signature on a document or sending the communication in the first place. In this context, non-repudiation refers to the ability to ensure that a party to a contract or a communication must accept the authenticity of their signature on a document or the sending of a message.

Accountability

Accountability is an essential part of an **information security plan**. The phrase means that every individual who works with an information system should have specific responsibilities for information assurance. One example would be a policy statement that all employees must avoid installing outside software on a company-owned information infrastructure.

6.7 Computer Protection

Computer protection or security is the process of protecting your computer against unauthorized intrusions. This unauthorized intrusion includes installation of malicious software on your computer without your consent and hackers taking over your computer's control so that it can be used as a botnet.

Computer Protection Basics

Once your PC security has been compromised, hackers can also take over your identity and use it to their advantage. Like, for example, hackers can pose as 'you' to carry out security attacks, can hide their actual location in case of high profile attacks and use yours instead, and can also encrypt your computer's data or crash it altogether. Some basic steps to protect a computer are:

1. **Install Antivirus Software:** Not just any antivirus software but one which can offer real-time protection. Because real-time protection is crucial if your PC is to be protected against today's security threats. Each day a new security threat is discovered and your antivirus software should keep up with them.
2. **Implement Other Security Tools:** Unfortunately, antivirus alone is not sufficient to avoid security threats. The whole package is required – anti-malware, firewall, and others – to combat today's sophisticated malware. These bunch of security tools usually are packaged and sold under the name Internet Security suite. PC users have to implement these if they need enhanced computer protection.
3. **Keep Your Computer Updated:** Ensure every software within your computer is properly patched up and up-to-date. This includes the operating system and the other third-party software which might be a part of it. Because most of the security threats exploit your system's security vulnerabilities and keeping your system up-to-date can eliminate such threats.
4. **Use Complex Passwords:** Another aspect which is very important but quite often ignored are strong and unguessable passwords. Each password you possess (not just your system password) to various accounts you might be having pose a serious threat to your PC and your digital identity.
5. **Follow Security Best Practices:** Finally, follow the security best practices like not clicking on suspicious mails, not downloading suspicious attachments, not visiting dangerous websites etc., to keep your computer and digital identity safe.

6.7.1 Antimalware vs Firewall

Firewall and antivirus are mechanisms which provide security to systems. Firewalls help control network traffic in the system by acting as barriers for incoming traffic.Whereas, antiviruses protect systems against internal attacks by perceiving or spotting malicious files and viruses.

Antivirus Software

Antivirus software is a cyber security mechanism. Its primary function is to scan, spot, and inhibit any distrustful files and software from getting into the system. Additionally, many traditional antivirus software incorporates detection of susceptibility, additional scanning of regular antivirus, and control of internet traffic.

Selection of Appropriate Antimalware?

Whether your computer or device runs Android, Mac or Windows, cybercriminals have sophisticated attacks to target machines across operating systems (OS). Choosing an antivirus software that's designed for your OS and tailored to your computing behaviours is important. Not all antivirus software is created equal, so there are following factors to consider:

1. Does your antivirus software protect you from major threats?
2. Does your program offer real-time prevention and rapid infection removal?
3. Does your software run accurately and efficiently?
4. Is your antivirus software easy to use?

6.7.2 Installing an antivirus program on a computer

To install an antivirus program on your computer, follow the steps below.

1. If you purchased the antivirus program from a retail store, insert the CD or DVD into the computer's disc drive. The installation process should start automatically, with a window opening to help guide you through the install process.
2. If you downloaded the antivirus program on the Internet, find the downloaded file on your computer. If the downloaded file is a zip file, unzip the file to extract and access the installation files. Look for a file named **setup.exe**, **install.exe**, or something similar, then double-click that file. The installation process should start, with a window opening to help guide you through the install process.
3. In the installation process window, follow the steps provided to install the antivirus program. The install process provides recommended options so the

antivirus program will function properly, which in most cases can be accepted as is.

4. When the install process is complete, close the install window.

The antivirus program is now installed and ready to use. It is recommended to restart your computer so that any modified settings in the operating system can take effect correctly.

Updating the antivirus Program

Without the latest definitions, the antivirus program will not know about the most recently created viruses and spyware, making your computer vulnerable to an infection. After installing the antivirus program, we highly recommend you update it with the latest virus and spyware definitions. The updates allow the antivirus program to protect your computer from all viruses and spyware. In many cases, the antivirus program automatically checks for and installs the latest updates. If prompted to do so, select Yes to update the antivirus program. If it does not prompt you to update immediately.

Performing a Scan

Correctly setting up and running an antivirus scan on your computer is one of the best starting defences for keeping your system free of malicious software. A strong defence begins with selecting an antivirus solution for your computer and understanding how to get the most out of it. Learning to properly use a virus scan will keep you much safer: without regular full scans, your system may be hiding some unpleasant surprises.

Virus scans search through your system to locate and remove any malicious threats on your device. You'll find most antivirus software guards against malware.

Follow these steps to run your first scan:

1. **Verify that all security updates are downloaded.** Updates will include all heuristic virus definitions and other data needed to scan properly.
2. **Execute a full virus scan.** This will scan hard drives, removable media, system memory, email and the like for viruses.
3. **Clean your backups first** by executing a virus scan before restoring any data.

4. **Read any result reports for further action.** Malware will usually be quarantined to await action from you. You'll have to decide what to do with the potential threat. Most antivirus apps give you a recommended action to simplify the decision for you.
5. **Follow prompts to remove or keep potential threats.** You'll want to remove most quarantined files usually. On occasion, you may find a clean file that has been miscategorized and given you a false positive. High-quality antivirus software should rarely label clean files as threats, so be sure you do your research on detection rates.

Teacher Notes:

- Perform an antivirus scan in class and discuss the results with students.

Fig. 6.1 Performing a Scan

6.8 Firewalls

A firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules. It establishs a barrier between secured and controlled internal networks that can be trusted and untrusted outside networks, such as the Internet. A firewall can be hardware, software, or both.

Top 10 Firewall Software

- FortiGate
- Check Point Next Generation Firewalls (NGFWs)
- Sophos XG Firewall
- Huawei Firewall
- Avast
- WatchGuard Network Security
- Palo Alto
- SonicWall
- GlassWire Firewall
- Cisco

Configuring a Firewall

Step 1: Secure your firewall

If an attacker is able to gain administrative access to your firewall it is "game over" for your network security. Therefore, securing your firewall is the first and most important step of this process. Never put a firewall into production that is not properly secured by at least the following configuration actions:

Update your firewall to the latest firmware.

- Delete, disable, or rename any default user accounts and change all default passwords. Make sure to use only complex and secure passwords.
- If multiple administrators will manage the firewall, create additional administrator accounts with limited privileges based on responsibilities. Never use shared user accounts.
- Disable simple network management protocol (SNMP) or configure it to use a secure community string.

Step 2: Architect your firewall zones and IP addresses

In order to protect the valuable assets on your network, you should first identify what the assets (for example, payment card data or patient data) are. Then plan out your network structure so that these assets can be grouped together and placed into networks (or zones) based on similar sensitivity level and function. For example, all of your servers that

provide services over the internet (web servers, email servers, virtual private network (VPN) servers, etc.) should be placed into a dedicated zone that will allow limited inbound traffic from the internet (this zone is often called a demilitarized zone or DMZ). Servers that should not be accessed directly from the internet, such as database servers, must be placed in internal server zones instead. Likewise, workstations, point of sale devices, and voice over Internet protocol (VOIP) systems can usually be placed in internal network zones. Generally speaking, the more zones you create, the more secure your network. But keep in mind that managing more zones requires additional time and resources, so you need to be careful when deciding how many network zones you want to use.

Step 3: Configure access control lists

Now that you have established your network zones and assigned them to interfaces, you should determine exactly which traffic needs to be able to flow into and out of each zone. This traffic will be permitted using firewall rules called access control lists (ACLs), which are applied to each interface or sub interface on the firewall. Make your ACLs specific to the exact source and/or destination IP addresses and port numbers whenever possible. At the end of every access control list, make sure there is a "deny all" rule to filter out all unapproved traffic. Apply both inbound and outbound ACLs to each interface and sub interface on your firewall so that only approved traffic is allowed into and out of each zone.

Step 4: Configure your other firewall services and logging

If your firewall is also capable of acting as a dynamic host configuration protocol (DHCP) server, network time protocol (NTP) server, intrusion prevention system (IPS), etc., then go ahead and configure the services you wish to use. Disable all the extra services that you don't intend to use. Configure your firewall to report to your logging server, and make sure that enough detailsare included.

Step 5: Test your firewall configuration

In a test environment, verify that your firewall works as intended. Don't forget to verify that your firewall is blocking traffic that should be blocked according to your

configurations. Testing your firewall should include both vulnerability scanning and penetration testing. Once you have finished testing your firewall, your firewall should be ready for production.

Logging Services in Firewall

A firewall, at its most basic form, is created to stop connections from suspicious networks. It inspects the source address (of data packet), destination address, and the destination port of all connections, and decides if a network can be trusted. For simplicity, we can aggregate information on the source address, source port, and destination address and port. The success of any firewall typically relies on the rules used to configure it. To be most effective, a firewall ruleset must be augmented with a successful logging feature. The logging feature documents how the firewall deals with traffic types. These logs offer insights into, for example, source and destination IP addresses, protocols, and port numbers.

Uses of Firewall

- To see if new firewall rules work well or to debug them if they do not work properly.
- To discover if any malicious activity is occurring within your network. However, it doesn't offer the information you need to isolate the source of the activity.
- If you identify repeated unsuccessful tries to access your firewall from a single IP address (or from a group of IP addresses), you may wish to create a rule to stop all connections from that IP.
- Outgoing connections derived from internal servers, for example, web servers, may show that someone is using your system as a launch pad. They could be launching attacks against computers on other networks from your system.

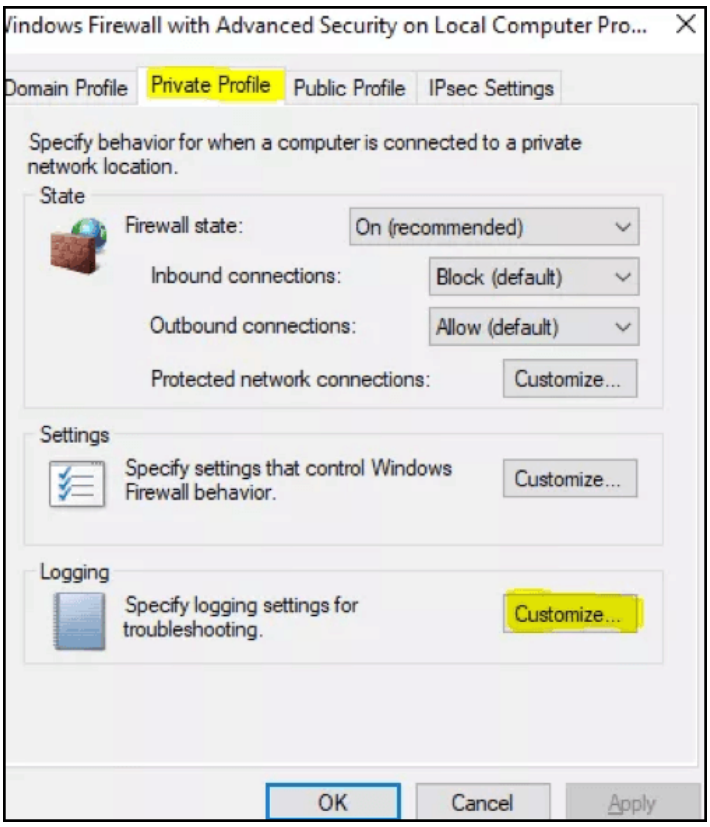


Fig. 6.2 Windows Firewall

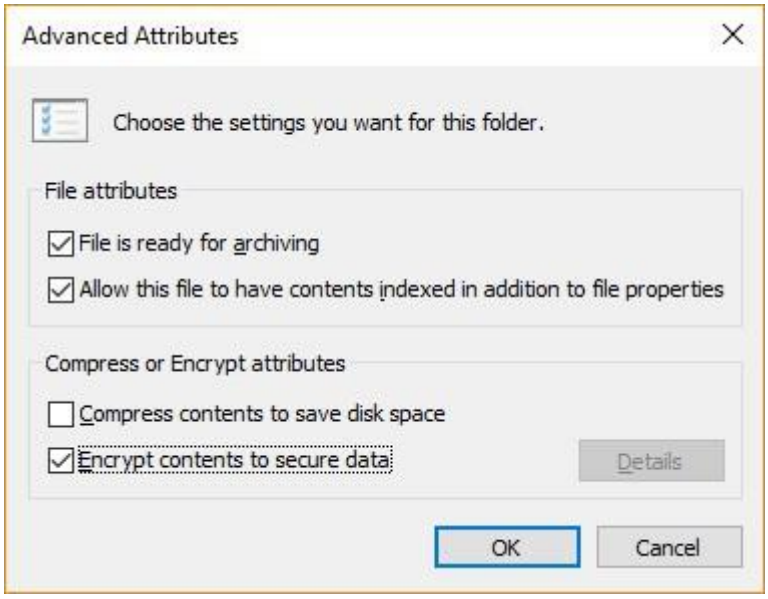
6.9 Encryption and Decryption

Encryption is the process of translating plain text data (*plaintext*) into something that appears to be random and meaningless (*ciphertext*). Decryption is the process of converting ciphertext back into plaintext.

Encrypting a file inWindows 10

File encryption helps protect your data by encrypting it. Only someone with the right encryption key (such as a password) can decrypt it. File encryption is not available in Windows 10 Home.

1. Right-click (or press and hold) a file or folder and select **Properties**.
2. Select the **Advanced** button and select the **Encrypt contents to secure data** check box.
3. Select **OK** to close the **Advanced Attributes** window, select **Apply**, and then select **OK**.



6.10 Backup and Restoration

Backup is the process of making a copy of information. The copy is typically transferred to a backup medium; i.e., either a virtual tape file, which is an allocation of disk space, or any physical backup device supported by Windows.Making backups of files on backup media is an important part of protecting data and ensuring the integrity of files. If a file is subsequently deleted by accident or if its data becomes corrupted, you can restore the most recent version of the file from backup media. Generally speaking, the system administrator at each site should establish routine backup procedures for the entire system. However, application programmers should know enough about these procedures to understand what is being backed up and with what frequency.

Creating Back up of Windows

There are several ways to back up your PC.

1. Select the **Start** button, then select **Control Panel** > **System and Maintenance** > **Backup and Restore**.
2. Do one of the following:
    - If you've never used Windows Backup before, or recently upgraded your version of Windows, select **Set up backup**, and then follow the steps in the wizard.

- If you've created a backup before, you can wait for your regularly scheduled backup to occur, or you can manually create a new backup by selecting **Back up now**.
- If you've created a backup before, but want to make a new, full backup rather than updating the old one, select **Create new**, **full backup**, and then follow the steps in the wizard.

Restore your files with File History

If you're missing an important file or folder that you've backed up, here's how to get it back:

1. In the search box on the taskbar, type **restore files**, and then select **Restore your files with File History**.
2. Look for the file you need, then use the arrows to see all its versions.
3. When you find the version you want, select **Restore** to save it in its original location. To save it in a different place, right-click **Restore**, select **Restore to**, and then choose a new location.

Find your backups created on previous versions of Windows

If you use *Backup and Restore* to back up files or create system image backups in previous versions of Windows, your old backup is still available in Windows 10. In the search box on the taskbar, type **control panel**. Then select **Control Panel** > **System and Security** > **Backup and Restore (Windows 7)**.

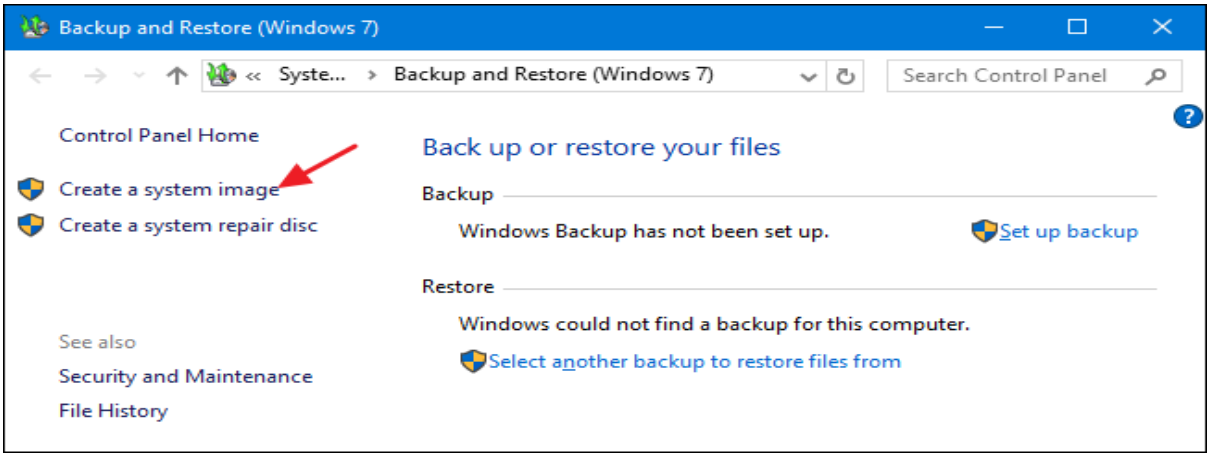


Fig. 6.3 Backup and Restore (Windows)

Key Points

- Computer security means dealing with the security of a standalone computer's software and hardware.
- Cyber security involves the security of individual computers plus the network they belong to and the intercommunication between all the computers in your organization, mobile devices, printers and more.
- Viruses are malicious programs that get loaded onto victims' computers without their knowledge or permission.
- Worms are software programs that will copy themselves from one computer to another, automatically.
- An attack vector is a path or means by which an attacker or hacker can gain access to a computer or network server in order to deliver a payload or malicious outcome.
- Hackers are people who use their knowledge for a good purpose and do not damage the data, whereas a cracker is someone who breaks into the system with a malicious purpose and damages data intentionally.
- Risk is the potential for loss, damage or destruction of assets or data caused by a cyber threat.
- Threat is a process that magnifies the likelihood of a negative event, such as the exploit of a vulnerability.
- A vulnerability is a weakness in your infrastructure, networks or applications that potentially exposes you to threats.
- Authentication is the process of identifying users that request access to a system, network, or device.
- Firewalls help control network traffic in the system by acting as barriers for incoming traffic.
- Antiviruses protect systems against internal attacks by perceiving or spotting malicious files and viruses.

Exercise

Select the most appropriate option

1. Which of the following is a component of an Information System?
   a. Hardware b. Software c. Telecommunication d. All of these

2. _____ are the people who use their knowledge for good purpose.

a. Intruders b. Hackers c. Crackers d. Both a and b

3. _____ is a weakness in your infrastructure.
   a. Risk b. Threat c. Vulnerability d. None of these

4. _____ controls network traffic.

a. Antivirus b. Firewall c. Wireshark d. Both a and b

5. ____ is a process of making a copy of information.

a. scanning b. restore c. backup d. Both a and b

Answer the following question.

1. Define Computer Security.
2. Differentiate between Computer Security and Cyber Security.
3. Define threats, attack vectors and vulnerability.
4. What are Information systems?
5. Describe assets of an Information system.
6. Define hacker, cracker and security expert..

7. Define confidentiality, integrity and authenticity.
8. Describe Encryption and Decyption.
9. Describe Malware.
10. Differentiate between Firewall and Anti-virus.

Practical Tasks

1. Install antimalware software on Windows.
2. Perform a malware scan on Windows.
3. Install and configure a firewall on Windows.
4. Configure firewall to create log files.
5. Perform a system restore on Windows.

Chapter 7

Communication Skills

After Studying this chapter, you will be able to

- know about communication process.
- understand types of communication (verbal/ non-verbal).
- understand effective communication skills.
- identify obstacles in communication.
- communicate effectively in a team environment.
- know about barrier of communication (cultural barriers, language barriers, gender barriers).
- create, access and manage email account.
- learn how to write and respond official email.
- understand the role of social media in marketing and business development.
- know the merits and de-merits of social media.

7.1 Communication

Communication is simply an act of transferring information from one place, person or group to another. Every communication involves (at least) one sender, a message and a recipient. This may sound simple, but communication is actually a very complex subject. The transmission of the message from sender to recipient can be affected by a huge range of factors. These include our emotions, the cultural situation, the medium used to communicate, and even our location.

7.1.1 Categories of Communication

There are various ways in which we communicate. The different categories of communication include:

- **Spoken or Verbal Communication**, which includes face-to-face, telephone, radio or television and other media.
- **Non-Verbal Communication**, covering body language, gestures, how we dress or act, where we stand, and even our scent. There are many subtle ways that we communicate (perhaps even unintentionally) with others. For example, the tone of voice can give clues to mood or emotional state, whilst hand signals or gestures can add to a spoken message.
- **Written Communication**, which includes letters, e-mails, social media, books, newspapers, magazines, the Internet and other media. Until recent times, a relatively small number of writers and publishers were very powerful when it came to communicating the written word. Today, we can all write and publish our ideas online, which has led to an explosion of information and communication possibilities.
- **Visualizations**: graphs and charts**,** maps, logos and other visualizations can all communicate messages.

7.1.2 Principles of Effective Communication

The following are the principles of effective communication:

- **Listening**: One of the most important aspects of effective communication is being a good listener.
- Non-Verbal Communication.
- Be clear and be concise.
- Be confident.
- Be empathetic.
- Always have an open mind.
- Convey respect.

7.1.3 Obstacles in Effective Communication

Barriers in communication can be categorized into four main types of constraints: Physical, psychological, organizational and semantic barriers. Physical barriers have nothing to do with the communicators themselves. Apart from this, there are also some cultural, language and gender barriers.

Besides physical and technical barriers, there are six barriers to effective communication:

Dissatisfaction or Disinterest with One's Job

If you are unhappy or have lost interest in your job, you are far less likely to communicate effectively. In other words, your heart isn't in it. This barrier, is perhaps the most difficult to overcome because it involves changing a mindset, and thus it typically doesn't change until the person leaves.

Inability to Listen to Others

Active listening is an important aspect of effective communication. You cannot engage with someone if you are not listening to them because you will tend to make assumptions about their needs based on your perceptions versus reality.

Lack of Transparency & Trust

It is extremely difficult to communicate anything when there is a lack of transparency and trust. For example, if your staff believes you are holding something back, they will be anxious, some will speculate, and as a result, it will be more difficult for them to process any attempt you make to communicate with them.

Communication Styles

Everyone has their own communication style. Some people are very direct while others prefer a more indirect approach. Some use detailed data, while others rely on generalities, and so forth. Occasionally, one person is so entrenched in their way of communicating, they find it difficult to communicate with others who rely on a different style. You might hear comments such as, “Mary never explains what she wants me to do, she’s never specific” or “Bill gets so caught up in the weeds, that I lose focus on the bigger picture.”

Conflicts in the Workplace

Conflict can happen for a variety of reasons and when it does, it becomes a barrier to effective communication. The nature of the conflict is not necessarily important, what is important is working to resolve the conflict. When conflict is not eradicated, it grows and then people begin to take sides, which further impedes effective communication.

Cultural Differences & Language

It is important to understand the cultural differences in communication. There can also be regional differences.For instance, a northerner might not like the term "y’all" or even understand the more comprehensive version, "all y’all.".

7.1.4 Principles for Effective Team Communication

Principles of effective communication are:

1. Have an open-door policy
2. Use project management software
3. Be open to feedback
4. Be clear about individual task responsibility
5. Do fun stuff to boost morale

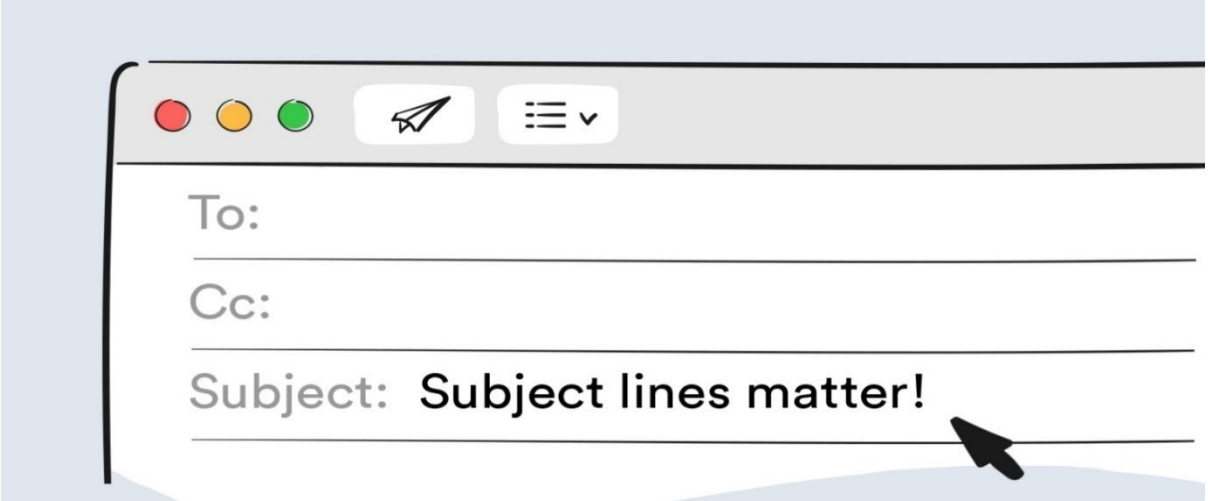
6. Give purpose to coffee breaks
7. Consider formal communications training
8. Decide what forms of communication work best
9. Identify group leaders
10. Understand diversity issues

7.2 Effective Email Writing

Email is a valuable tool but in the meantime, it creates some challenges for writers. Miscommunication can easily occur when people have different expectations about the messages that they send and receive. Email is used for many different purposes, including contacting friends, communicating with professors and supervisors, requesting information, and applying for jobs, internships, and scholarships. Depending on your purposes, the messages you send will differ in their formality, intended audience, and desired outcomes. The following are some important components of an email:

Subject Lines

Email subject lines are like newspaper headlines. They should convey the main point of your message or the idea that you want the reader to take away. Therefore, be as specific as possible. One-word subjects such as “Hi,” “Question,” or “FYI” are not informative and don’t give the reader an idea of how important your message is. If your message is time sensitive, you might want to include a date in your subject line, like “Meeting on Thurs, Dec 2”.



Note for teachers.

- Compose an example email in class.

Greetings and Sign-offs

Use some kind of greeting and some kind of sign-off. Don't just start with your text, and don't just stop at the end without a polite signature. If you don't know the person well, you may be confused about how to address them ("What do I call my TA/professor?") or how to sign off (Best? Sincerely?). Nonetheless, it is always better to make some kind of effort. When in doubt, address someone more formally to avoid offending them. Some common ways to address your reader are:

- Dear Professor Smith,
- Hello Ms. McMahon,
- Hi Mary Jane,

If you don't know the name of the person you are addressing, or if the email addresses a diverse group, try something generic, yet polite:

- To whom it may concern,
- Dear members of the selection committee,
- Hello everyone,

Your closing is extremely important because it lets the reader know who is contacting them. Always sign off with your name at the end of your message. If you don't know the reader well, you might also consider including your title and the organization you belong to; for example:

Mary Watkins
Senior Research Associate
Bain and Company

Joseph Smith
UNC-CH, Class of 2009

For your closing, something brief but friendly, or perhaps just your name, will do for most correspondence:

- Thank you,
- Best wishes,
- See you tomorrow,
- Regards,

For a very formal message, such as a job application, use the kind of closing that you might see in a business letter:

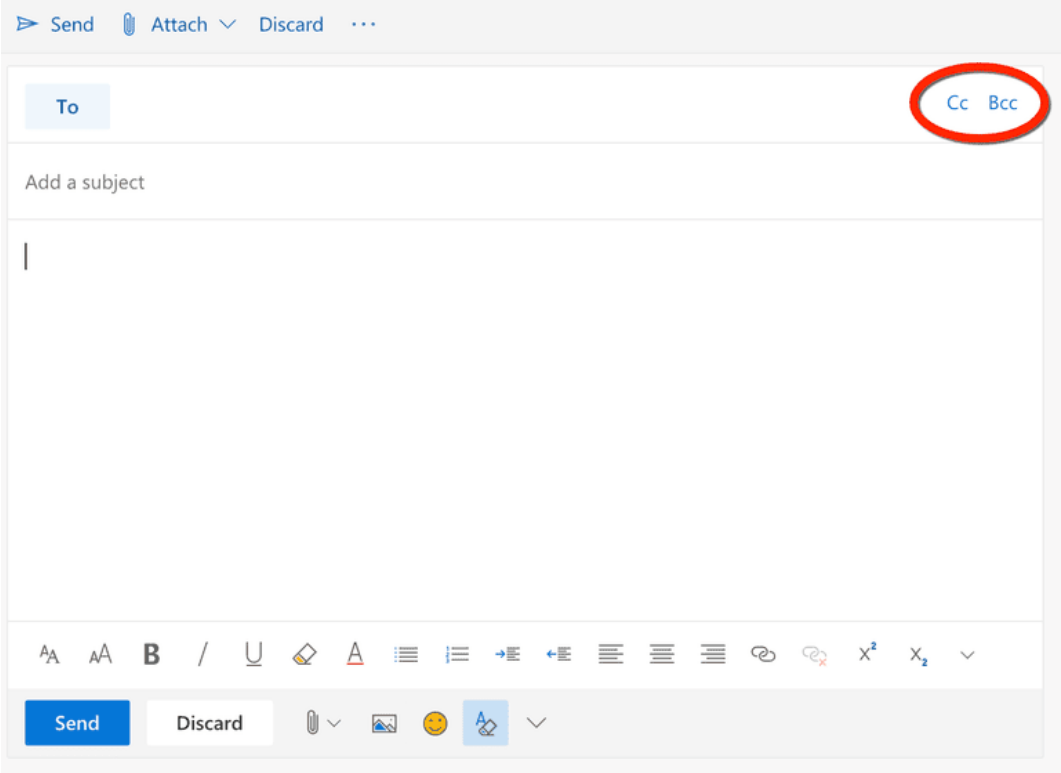
- Sincerely,
- Respectfully yours,

Cc: and Bcc: ('carbon copy' and 'blind carbon copy')

Copying individuals on an email is a good way to send your message to the main recipient while also sending someone else a copy at the same time. This can be useful if you want to convey the same exact message to more than one person. In professional settings, copying someone else on an email can help get things done, especially if the person receiving the copy is in a supervisory role. For example, copying your boss on an email to a non-responsive co-worker might prompt the co-worker to respond. Be aware, however, that when you send a message to more than one address using the Cc: field, both the original recipient and all the recipients of the carbon copies can see all the

addresses in the To: and Cc: fields. Each person who receives the message will be able to see the addresses of everyone else who received it.

Blind copying emails to a group of people can be useful when you don't want everyone on the list to have each other's addresses. The only recipient address that will be visible to all recipients is the one in the To: field. If you don't want any of the recipients to see the email addresses in the list, you can put your own address in the To: field and use Bcc: exclusively to address your message to others. However, do not assume that blind copying will always keep recipients from knowing who else was copied—someone who is blind copied may hit "reply all" and send a reply to everyone, revealing that they were included in the original message.



7.3 Role of Social Media in Marketing

Marketing is a tool to inform consumers about your products, who you are, and what you offer. Social media accomplishes all of these goals:

- You can use social media to define your brand identity and the products or services that you offer.
- Social media allows you to create relationships with people who might not know about your products and services or what your company represents.
- Social media can make you "real" to consumers. If you want people to follow you, don't just talk about the latest product news, but share your personality with them.
- You can use social media to associate yourself with peers that may be serving the same target market.

Advantages of Social Media

The business benefits of effective social media use include:

- **Brand awareness** - Compelling and relevant content will grab the attention of potential customers and increase brand visibility.
- **Brand reputation** - You can respond instantly to industry developments and be seen as 'thought leader' or expert in your field. This can improve how your business is seen by your audience.
- **Cost effective** - It can be much **cheaper** than traditional advertising and promotional activities. The costs of maintaining a social media presence are minimal. If you choose to invest in paid advertising, you can spend as much or as little as your budget allows.
- **Website traffic** - Social content can boost traffic to your website. This can lead to increased online conversions such as sales and leads.
- **Evaluation** - It is easy to measure how much website traffic you receive from social media. You can set up tracking to determine how many sales are generated by paid social advertising.
- **Customer interaction** - You can deliver improved customer service and respond effectively to feedback. Positive feedback is public and can be persuasive to other potential customers. Negative feedback highlights areas where you can improve.

- **Target audience -** Customers can find you through the social media platforms they use most. You can choose to maintain a presence on particular platforms that are in line with your target audience. For example, if you are targeting young people, you could reach them through Snapchat. If you are seeking business customers, LinkedIn may be the best channel.

Disadvantages of Social Media

Some of the disadvantages of using social media include:

- **Resources**: Resources are required to manage presence on social media i.e., responding to feedback and producing new content. This can include hiring and training staff, investing in paid advertising and paying for the costs of creating video or image content.
- **Ineffective use**: Social media can be used ineffectively. For example, using social media to push for sales without engaging with customers, or failing to respond to negative feedback - may damage your reputation.

Key Points

- Communication is simply the act of transferring information from one place, person or group to another.
- These barriers can be categorized into four main types of constraints to effective communication: Physical, psychological, organizational, as well as semantic barriers.
- Email subject lines are like newspaper headlines. They should convey the main point of your message or the idea that you want the reader to take away.
- Copying individuals on an email is a good way to send your message to the main recipient while also sending someone else a copy at the same time.
- Blind copying emails to a group of people can be useful when you don't want everyone on the list to have each other's addresses.
- Marketing is a tool we use to inform consumers about our products, who we are, and what we offer.

Exercise

Select the most appropriate option.

1. Which opion hides other recipients from all recipients.
   a. CC b. BCC c. Subject Line d. All of these

2. Communication barriers can be categorized into _____ types.

a. 5 b. 3 c. 4 d. Both a and b

3. The following quality of your busness builds relationships.
   a. Brand loyalty b. Customer Interaction c. Evaluation d. None of these

4. _____ contains the headlines of your email.

a. Attachment b. Subject line c. Text d. Both a and b

5. ____ is a face to face medium of communication.

a. Spoken b. Verbal c. Non verbal d. Both a and b

Answer the following question.

1. Define Communication Skills.
2. Whats are the different forms of communication?
3. What are the principles for an effective communication?
4. What are the common barriers in communication?
5. What are themain components of an email?
6. What communication skills are required in a team work?
7. Describe advantages and disadvantages of social media.

Glossary

ADSL: Asymmetric digital subscriber line (ADSL) is a type of digital subscriber line (DSL) technology, a data communications technology that enables faster data transmission over copper telephone lines than a conventional voiceband modem can provide.

AI: Artificial intelligence (AI) is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence.

Algorithm: A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

ANSI: The American National Standards Institute - ANSI - facilitates and coordinates the U.S. voluntary standards and conformity assessment system.

Arithmetic: Arithmetic is the branch of mathematics that deals with the study of numbers using various operations on them

BIOS:BIOS (basic input/output system) is the program a computer's microprocessor uses to start the computer system after it is powered on.

Cyber: relating to, or involving computers or computer networks (such as the Internet).

Data Warehouse: In computing, a data warehouse (DW or DWH), also known as an enterprise data warehouse (EDW), is a system used for reporting and data analysis and is considered a core component of business intelligence.

DNS: The Domain Name System (DNS) is the phonebook of the Internet.

EEPROM: EEPROM (electrically erasable programmable read-only memory) is user-modifiable read-only memory (ROM) that allow users to erase and reprogram stored data repeatedly in an application.

Extension: A filename extension, file extension or file type is an identifier specified as a suffix to the name of a computer file.

Floating Point: In computing, floating-point arithmetic (FP) is arithmetic using formulaic representation of real numbers as an approximation to support a trade-off between range and precision.

FORTRAN: FORTRAN is a general-purpose, compiled imperative programming language that is especially suited to numeric computation and scientific computing.

Gateway: A gateway is a node (router) in a computer network, a key stopping point for data on its way to or from other networks.

Geographic: based on or derived from the physical features of an area.

Hotspot: A hotspot is a physical location where people can access the Internet, typically using Wi-Fi, via a wireless local area network (WLAN) with a router connected to an Internet service provider.

Java: Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

Logistics: the commercial activity of transporting goods to customers.

Machine Learning: Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data.

Network Traffic: Network traffic is the amount of data moving across a computer network at any given time.

NFC: Near-Field Communication (NFC) is a set of communication protocols for communication between two electronic devices over a distance of 4 cm.

Off-site:not located or occurring at the site of a particular activity.

Precedence: the condition of being considered more important than someone or something else; priority in importance, order, or rank.

Python: Python is a powerful general-purpose programming language. It is used in web development, data science, creating software prototypes, and so on.

ROM: ROM is an acronym for Read-Only Memory. It refers to computer memory chips containing permanent or semi-permanent data.

Topology: Network topology is the arrangement of the elements (links, nodes, etc.) of a communication network.

Troubleshoot: Analyse and solve serious problems for a company or other organization.

UEFI: The Unified Extensible Firmware Interface (UEFI) is a publicly available specification that defines a software interface between an operating system and platform firmware.

Unary: In mathematics, a unary operation is an operation with only one operand, i.e. a single input.

UNIX: UNIX is an operating system which was first developed in the 1960s, and has been under constant development ever since.

About the Author

Muhammad Umair is a researcher at the *Sensors, Cloud and Services (SCS) Lab*, School of Computer Sciences, The University of Sydney, Australia. Muhammad Umairis also a Lecturer at the Department of Electrical, Electronics and TelecommunicationEngineering, New Campus, UET Lahore. He completed his B.Sc. Electrical Engineering and M.Sc.Electrical Engineering from University of Engineering & Technology (UET) Lahore in 2014 and 2017,respectively. He has worked as a Research Officer at Internet of Things (IoT) lab at Al-KhwarizmiInstitute of Computer Sciences, UET Lahore. He has also worked at Sultan Qaboos IT Researchlab as a Research Officer. His survey on Social IoT platforms is the most cited survey for SIoTapplications. He has designed graduate level courses on IoT.

قومی ترانہ

پاک سَر زمین شاد باد! کشورِ حسین شاد باد!
تو نِشانِ عزمِ عالی شان ارضِ پاکستان
مرکزِ یقین شاد باد!

پاک سر زمین کا نِظام قُوّتِ اُخوّتِ عوام
قوم ، مُلک ، سلطنت پائندہ تابندہ باد!
شاد باد منزلِ مُراد!

پرچَمِ سِتارہ و ہِلال رہبرِ ترقی و کمال
ترجُمانِ ماضی، شانِ حال جانِ اِستقبال
سایۂ خُدائے ذُوالجلال!

National Vocational & Technical Training Commission (NAVTTC)
Plot No.38, Sector H-9/4, Kirthar Road, Islamabad.
Tel: +92-51-9207518
Website: www.navttc.gov.pk