Lesson 5.1.1 – Understanding Computer Components

Getting Started: Why Learn About Computer Components?

Before you can protect or troubleshoot a computer, you need to understand how it works. Every computing device — from a smartphone to a powerful desktop — is made up of interconnected components that process, store, and transmit data. Once you understand what each part does, you can begin to see how problems occur — whether through hardware failure, software bugs, or security vulnerabilities.

The Necessary Components of a Computer

A computer is an electronic device that takes input, processes data, stores information, and produces output. To perform these functions, certain hardware components are absolutely necessary, while others are optional but enhance performance or usability.

Required Components

- **Processor (CPU)** – The *Central Processing Unit* is the brain of the computer. It performs calculations, runs programs, and manages all other components.
- **Memory (RAM)** – *Random Access Memory* temporarily stores data and instructions that the CPU needs right now. It's fast but *volatile*, meaning everything in RAM is lost when the computer is turned off.
- **Motherboard** – The central circuit board that connects all components. It allows communication between the CPU, memory, storage devices, and input/output ports.
- **Power Supply** – Converts electrical power from the wall outlet into usable energy for the computer's components.

Optional (But Helpful) Components

- **Permanent Storage (Hard Drive or SSD)** – Used to store the operating system, applications, and files even when the power is off.
- **Graphics Card (GPU)** – Handles rendering of images, video, and animation. Essential for gaming, design, or AI applications.
- **Network Interface Card (NIC)** – Enables network or internet connections.
- **Display Screen** – Allows users to see the computer's output.

These components work together in nearly every device — whether it's a Raspberry Pi, smartphone, or MacBook — though their size, power, and purpose vary greatly.

Input: How Data Gets Into the Computer

Computers rely on *input devices* to receive data and commands from users or the environment. Common examples include:

- **Keyboard and Mouse** – Standard input devices for typing and navigation.
- **Voice Input** – Microphones or voice assistants like Siri or Alexa.

Input: How Data Gets Into the Computer Continued…

- **Cameras and Video Devices** – Capture images and motion data.
- **Touchscreens** – Combine input and output by allowing users to interact directly with the display.
- **Game Controllers or Sensors** – Translate physical motion into digital input.
- **Network Connections** – Bring in data from other devices or the internet.

Security Risk Example: Wireless Keyboard Sniffing

A *wireless keyboard sniffer* can intercept signals between a keyboard and its receiver, capturing keystrokes (including passwords). This highlights why understanding *how data enters* a system is vital for cybersecurity.

Processing: The Role of the CPU

The Central Processing Unit (CPU) is the “brain” of the computer. It interprets and executes instructions from software. Each CPU performs billions of operations per second.

CPU Types

- **x86 Architecture** – Made by Intel and AMD, found in most PCs, Macs, and Chromebooks.
- **ARM Architecture** – Found in smartphones, tablets, and devices like the Raspberry Pi. ARM chips are becoming more common in modern laptops because they use less power and generate less heat.

A 32-bit CPU can handle smaller amounts of data at once, while a 64-bit CPU can process more data simultaneously and access much larger amounts of memory. The architecture determines what type of software your system can run and how efficiently it performs tasks.

Security Risks: Spectre, Meltdown, and Foreshadow

These are famous hardware vulnerabilities discovered in CPUs. They exploit how processors temporarily store data in memory. Attackers can use these flaws to access private information stored by other programs. It’s a reminder that even the most powerful components can have weaknesses.

Memory: The Speed of RAM

RAM (Random Access Memory) temporarily holds data that's currently being used by the computer. It allows quick access for running programs and multitasking. Because it's *volatile*, all data in RAM disappears when power is lost.

- **Speed** Comparison:
  - RAM transfers data in nanoseconds.
  - Hard drives transfer data in milliseconds.
    This makes RAM thousands of times faster than hard drives.

What Can Go Wrong: Cold Boot Attack

If a computer is shut down improperly, remnants of data can sometimes be retrieved from RAM before it completely loses power. Attackers can use a *cold boot attack* to extract sensitive information like passwords or encryption keys.

Storage: Holding Your Data

Unlike RAM, storage devices retain information even after the computer is turned off.

Types of Storage

- **Hard Disk Drive (HDD)** – Uses spinning disks and magnetic storage. Slower, but cheaper and can store more data.
- **Solid-State Drive (SSD)** – Uses flash memory with no moving parts. Much faster and more reliable, but typically more expensive.

Removable Storage

USB drives, memory cards, and external hard drives allow users to transfer data between devices. However, they can also introduce malware if not properly scanned.

Output: How Data Leaves the Computer

Output devices present data to users in visual, auditory, or physical form.

Examples

- **Display Screen or Monitor** – Shows text, graphics, and video.
- **Speakers or Headphones** – Provide sound output.
- **Printers** – Create physical copies of documents or images.
- **Vibration Motors or VR Goggles** – Offer sensory feedback in games or simulations.
- **GPS Displays** – Provide navigation and mapping information.

Security Risk Example: “A Monitor Darkly” Attack

This refers to exploiting vulnerabilities in how screens display data — for example, inserting malicious code into video drivers or display firmware that could capture or distort screen output.

From Human to Machine Language: Decoding Instructions

Computers don’t understand English, Spanish, or even basic math symbols. They only understand **binary code**, made up of **0s and 1s** — representing “off” and “on” electrical signals.

How Humans Communicate with Machines

1. **High-Level Languages:** Programmers write code in human-friendly languages like Python, Java, or C++.
2. **Compiler or Interpreter:** Converts that code into **machine code** (binary) so the CPU can execute it.
3. **Execution:** The CPU processes those binary instructions and performs the requested tasks.

This translation process bridges the gap between human logic and machine precision.

From Hardware to Software

Hardware is the physical part of a computer. Software is the collection of programs and instructions that tell the hardware what to do.

Operating Systems

The *operating system (OS)* manages all hardware and software resources. Examples include:

- **Windows**
- **macOS**
- **Linux**
- **Android**
- **iOS**

The OS handles file management, device communication, and user interfaces. It’s the middle layer that allows software applications to communicate with the hardware without the user needing to know binary or assembly code.

In Summary

Computers are made up of interconnected parts that work together in a precise, logical flow:

Input → Processing → Storage → Output

Knowing how data moves through this process — and where weaknesses can appear — is the first step in protecting, repairing, and mastering computer technology.

Lesson 5.2.1 – Network Components

A computer network isn't just a bunch of machines connected with cables — it's a system designed to share information, resources, and services.

To build one, we need four main ingredients:

1. Computers (Hosts):

These are the devices that want to share information. Examples include desktops, laptops, servers, and even smartphones. In networking terms, these are called hosts because they "host" data and applications that others can use.

2. Media (Cables or Air):

Every network needs a way to move data. This can be wired media (like Ethernet cables made of copper or fiber optic strands) or wireless media (like Wi-Fi signals that travel through the air).

3. Network Devices (Hubs, Switches, Routers):

These devices are like the traffic directors of a network. They make sure data goes to the right destination, whether that's another computer down the hall or across the world.

4. Things to Share:

Peripherals: Devices such as printers, scanners, or copiers that can be shared by multiple users.

Services: Shared digital tools such as file storage, websites, and email systems that allow collaboration and communication.

A **network** is a group of two or more computers connected so that users can access the same data, resources, and applications at the same time.

While a two-computer setup counts as a network, most modern networks have dozens, hundreds, or even thousands of connected devices.

A Local Area Network (LAN) connects devices within a limited geographic area — such as a home, school, or business office.

LANs make it possible to share:

- Files and folders
- Printers and other peripherals
- Software applications

LANs range in size from two connected computers to thousands of devices across a large campus. To work properly, all devices in a LAN must share a common IP address range, which is like having matching postal codes for easy delivery of data.

WAN – Wide Area Network

A Wide Area Network (WAN) connects devices and LANs across long distances. WANs can span cities, states, or even continents. In most organizations, WANs link multiple office locations together. The

Internet is the world's largest WAN — a vast system of interconnected networks represented by the familiar "cloud" icon in diagrams. Where LANs are like local neighborhoods, WANs are like global highways that connect them all.

Connection Devices

Different devices play unique roles in moving information through a network:

Hub

- A hub sends every message it receives to all connected devices.
- This is called a broadcast — similar to yelling a message in a crowded room.
- Because hubs don't filter or direct traffic, they are considered "dumb" devices.
- They simply repeat signals without knowing where they should go.

Switch

- A switch is smarter. It learns which device is connected to which port and sends messages only to the correct destination.
- This reduces unnecessary traffic and improves network performance.
- A switch uses MAC (Media Access Control) addresses to identify and communicate with devices directly — like making a local phone call to one specific person.
- A Wireless Access Point (WAP) works like a switch but without wires, connecting wireless devices to the network.

Router

- A router connects multiple networks together — it's the only device that can do so.
- Routers decide the best path for data to travel between devices on different networks (for example, from your home LAN to a website's server on the Internet).
- Routers can connect local networks (LANs) to wide networks (WANs), making them essential for Internet access.
- Because routers connect networks, they are sometimes called gateways.

Example: Imagine the Accounting Department has its own network, and the Marketing Department has another. A router allows them to share information between the two safely and efficiently. Routers "talk" to each other, keeping track of the paths data must take to reach its destination anywhere in the world.

How Packets Travel

When one computer (say, PC1) wants to send a message to another (PC4), the process depends on the network device:

Through a Hub: The hub doesn't know who the message is for, so it sends it to everyone. Every device receives it, but only PC4 will open it. This wastes bandwidth.

Through a Switch: The switch already knows which port PC4 is on and sends the message directly there. This makes communication faster and more secure.

Lesson 5.2.2 – Network Naming

In simple terms, networking is all about **sharing information** between devices. Just like humans have to follow certain social rules when talking or writing, computers follow **networking rules** to communicate smoothly and efficiently.

Think about how humans communicate:

Speaking: in person, on the phone, or in a group. We follow rules like taking turns, raising our hands, and not shouting over others.

Writing: through letters, emails, or text messages. These forms have their own conventions—addressing the right person, using salutations, and proper punctuation.

Computers do the same thing when "talking" to each other across a network. They follow structured rules called protocols that determine:

1. How to find other devices on the network

2. How to establish and maintain a connection

3. How to send, receive, and interpret different types of information

Without these rules, communication would be chaotic, and devices wouldn't know who's talking to whom.

Rule #1 – Every Device Must Be Unique (The MAC Address)

Every device that connects to a network must have a **unique identity** so it can be recognized. That unique ID is called a MAC Address, which stands for Media Access Control Address. A MAC address is tied to the Network Interface Card (NIC) — the component that allows a device to connect to the network. Each NIC has its own permanent, factory-assigned MAC address. If a computer has multiple network connections (like Wi-Fi and Ethernet), each one has a different MAC address.

The MAC address acts as the physical address of a device. It is represented as 12 hexadecimal digits, often separated by colons or hyphens: '00:24:E8:83:68:96` or `00-C0-CA-52-38-8C`

It's similar to a Social Security Number — unique, unchangeable (in theory), and specific to that one device. However, there's one catch: a MAC address can be spoofed, meaning someone can disguise their device to look like another by faking its MAC address. This is often used in cybersecurity testing or, unfortunately, in attacks to hide identity.

Rule #2 – You Must "Belong" to a Network (The IP Address)

Once a device is physically connected, it needs permission to belong to a specific network. This is done through an IP Address (Internet Protocol Address). An IP address is like a temporary, logical address that identifies a device while it's connected. It allows a computer to send and receive data across the network or the Internet.

When you join a network and get an IP address, you gain access to:

- Shared files and folders
- Networked devices like printers and scanners
- The Internet and cloud-based services

An IP address works a lot like an ID card:

- It can change over time (for example, every time you reconnect to Wi-Fi)
- It's unique within that particular network
- It can be reassigned by the network automatically using something called DHCP (Dynamic Host Configuration Protocol)

There are two main types of IP addresses:

1. IPv4 (Internet Protocol version 4):

The older format, but still the most common. It consists of four numbers separated by dots, such as `192.168.55.32`. Each section is called an **octet** and can range from 0 to 255.

2. IPv6 (Internet Protocol version 6):

A newer format developed because the world is running out of IPv4 addresses. IPv6 uses longer, more complex combinations of letters and numbers, like `2001:0db8:85a3:0000:0000:8a2e:0370:7334`. IPv6 allows trillions more devices to have unique addresses, which is essential for the Internet of Things (IoT) era.

Rule #3 – Devices Must Know Who They're Talking To (ARP)

For successful communication, a sending device must know both the IP address (logical location) and the MAC address (physical location) of the destination device. To match these two pieces of information, networks use a process called ARP — the Address Resolution Protocol.

Here's how ARP works:

1. When a device wants to send data, it first checks if it knows the MAC address that goes with the target IP address.
2. If it doesn't, it sends an **ARP request** — a broadcast message asking, "Hey, who has IP address 10.0.0.1?"
3. The device with that IP address responds with its MAC address.
4. The sender then saves this information in an **ARP table** for future use.

Example:

- Bob's computer at `10.0.0.3` sends a message: "Who is 10.0.0.1?"
- Sally's computer replies: "That's me! My MAC address is A1:FF:32:5A:EC:AA."
- Bob stores that information so next time, he can contact Sally directly without broadcasting again.

This process happens constantly, behind the scenes, every time computers talk on a local network.

Lesson 5.3.2 – Packet Delivery and Protocols

Before modern computer networks, communication systems worked very differently.
Traditional landline telephones used something called circuit switching, which means a dedicated communication line was created between two people for the duration of the call. No one else could use that connection until the call ended.

Think of it like sending a letter through the mail:

- You write the entire message first.
- You seal it in an envelope.
- It travels along one specific route.
- The receiver opens it and reads the whole message from start to finish.

This method works well for conversations or letters, but it isn't efficient for the high-speed, data-heavy communication that happens on the Internet today.

Digital Packets – The Modern Way

Modern digital networks use **packet switching**, which is faster and more flexible.
Instead of sending one large continuous message, the data is broken up into **packets** — small pieces of information that can travel independently through the network.

Each packet contains:

- A piece of the original message
- The sender's and receiver's addresses
- A sequence number (to help put the pieces back in order)
- Error-checking information

These packets might travel **different routes** to reach the same destination. Some may arrive quickly, others slowly, and sometimes out of order — but once they all arrive, the receiver's computer reassembles them into the original message

Imagine writing a long story but instead of mailing it as one letter, you split it into ten **postcards**, each numbered from 1 to 10.
Each postcard might take a different route through the postal system — some could arrive early, others late, and maybe not in order.

When your friend receives them, they use the numbers to:

1. Reorder the postcards correctly
2. Check if any are missing
3. Read the full story once everything is complete

That's exactly how packet switching works on the Internet. It's fast, efficient, and able to handle millions of users communicating at once.

Protocols – The Rules of Communication

All this organized chaos requires structure. The rules that govern how data is sent, received, and processed are called protocols. The most important family of these rules is known as the TCP/IP suite.

- It includes over 100 individual protocols.
- Each defines a specific rule for how a certain type of data should be handled.
- It's named after the two main protocols: TCP (Transmission Control Protocol) and IP (Internet Protocol).

Think of TCP/IP as a set of traffic laws for the Internet — making sure data doesn't get lost or confused while traveling between devices.

Protocols and Ports

Every type of network service (like a website or email system) uses a specific port number — a virtual "door" that determines which application should handle incoming data.

When a computer receives data, it looks at the port number to decide what to do with it:

- **Port 80** → Web browsing (HTTP)
- **Port 25** → Email (SMTP)
- **Port 22** → Secure connections (SSH)

If a server isn't using the standard port number for a service, users have to be told which "door" to knock on instead.

Physical vs Virtual Ports

- Physical Ports: These are actual hardware connectors like USB, Ethernet, or HDMI ports — the ones you can plug cables into.
- Virtual Ports: These are software-based "channels" used to manage network communication. They are numbered from 0 to 65,535 and ensure that multiple conversations can happen on one device without getting mixed up.

Common Protocols and Their Port Numbers

Here are some of the most common examples you'll see in networking:

Category	**Protocol**	**Port(s)**	**Purpose**
File Transfer	FTP	20, 21	Transfer files between computers
	SFTP	115	Secure file transfer
Remote Access	SSH	22	Secure remote login
	Telnet	23	Basic remote connection

Category	Protocol	Port(s)	Purpose
	RDP	3389	Remote Desktop access
Network Management	DNS	53	Translates domain names into IP addresses
	DHCP	67	Assigns IP addresses automatically
Email	SMTP	25	Sends email
	POP3	110	Receives email
	IMAP	143	Accesses email remotely
Web Browsing	HTTP	80	Standard web traffic
	HTTPS	443	Secure web traffic (encrypted)

Every packet sent across the Internet must use one of two transport protocols: TCP or UDP.

TCP – Transmission Control Protocol

- Connection-oriented: establishes a link before sending data (like a phone call).
- Reliable: tracks packets, ensures none are lost, and reorders them correctly.
- Resource-intensive: uses more time and system power but guarantees accuracy.
- Common uses: web browsing, email, file downloads, and most online games.

TCP uses a process called a 3-Way Handshake:

1. Device A says, “Can we talk?”
2. Device B replies, “Yes, I hear you.”
3. Device A confirms, “Great — let’s begin!”

Only then does data start flowing, ensuring both sides are ready.

UDP – User Datagram Protocol

- Connectionless: sends packets without setting up a session first.
- Fast but unreliable: no guarantee all packets arrive or arrive in order.
- Lightweight: great for speed-critical activities where missing a few packets doesn’t matter much.
- Common uses: streaming music or video, VoIP (Internet phone calls), DNS, and fast-paced games.

Think of UDP as yelling short messages quickly — some might get lost, but the conversation keeps moving fast.