Level Up

Through **Digital Discoveries** 10

McGraw Hill

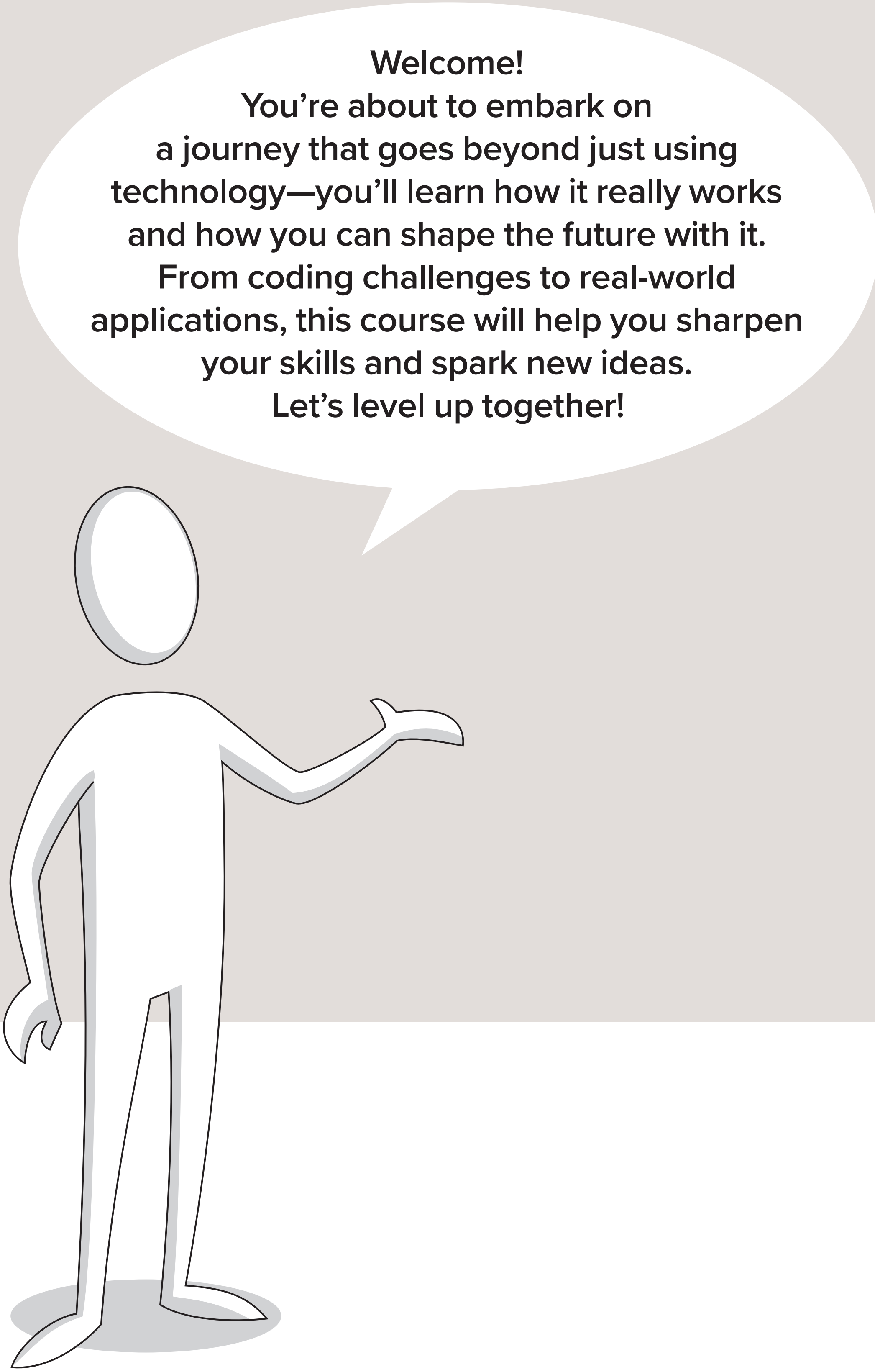
binarylogic

Level Up

Through **Digital Discoveries**

10

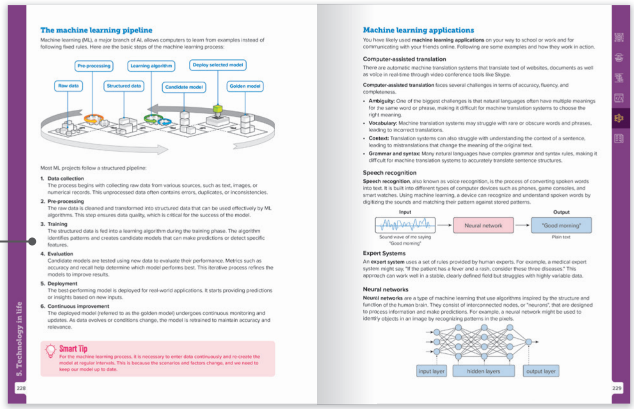
Contents


Welcome!
You're about to embark on a journey that goes beyond just using technology—you'll learn how it really works and how you can shape the future with it. From coding challenges to real-world applications, this course will help you sharpen your skills and spark new ideas. Let's level up together!

Key Features

An innovative approach to building digital competencies, developed by expert educators.

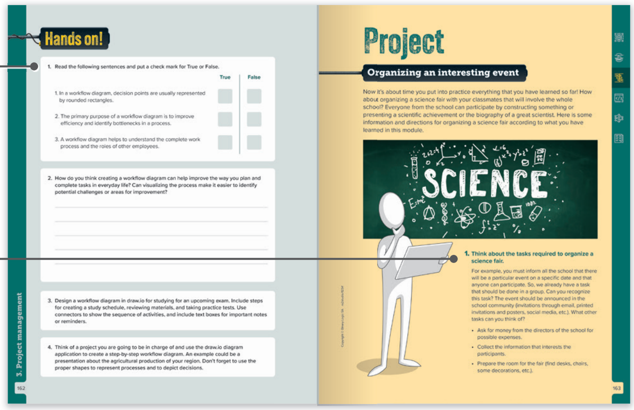
Each unit offers straightforward explanations and contemporary examples, making technology concepts accessible and relevant.



Curriculum aligns with the latest industry standards, preparing students for certifications and future careers.

Every unit includes a variety of tasks and activities designed to help students build essential digital competencies.

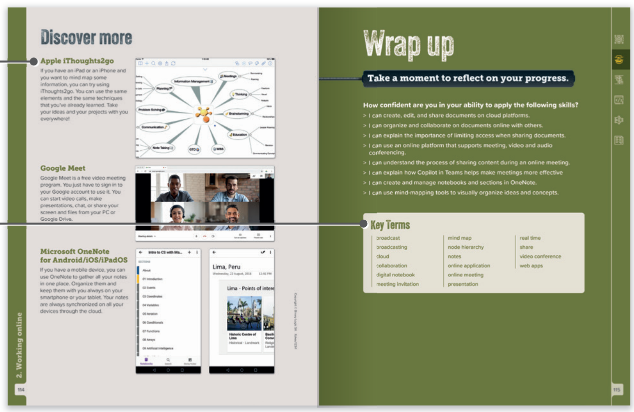
Projects and exercises throughout the course reinforce students' understanding and practical application of digital skills in real-world scenarios.



Well-defined learning goals and hands-on, applicable digital skills.

Students learn about platform diversity, expanding their digital toolkit and adaptability.

Each unit organizes key terms that are crucial for digital literacy, equipping students for today's technology-driven workplace.



1. Foundations of computer science

Understanding how computers work is important because they play a significant role in daily life, including activities such as homework, entertainment, and communication. This unit covers how computers store and process data, how they communicate over networks, and how key components like logic gates and operating systems contribute to their efficient operation.

Learning Objectives

In this unit, you will:

> identify the difference between decimal, binary, and hexadecimal systems, and how to convert between them.
> represent numbers, text, and images in binary using the ASCII table.
> use Boolean logic and logic gates to process binary data.
> understand the role of transistors and integrated circuits in computers.
> understand how abstractions are applied in daily life.
> understand what multiprogramming is and how operating systems manage tasks.
> illustrate data transmission through packet switching.
> understand the role of IP addresses and hostnames in communication.
> outline the functions of each layer in the OSI model and their impact on networking.
> understand the importance of firewalls and setting them up.

Tools

> Microsoft Windows

LESSON 1

Data manipulation

What language do you think a computer uses to perform all its tasks? How do computers represent different types of data, like text, images, and numbers?

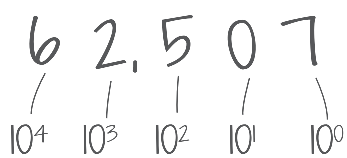
Modern digital computers are virtually everywhere around us. We use desktop computers at home, school, and work; powerful laptops that can be easily transported; and smartphones, fully functional computers that fit in our pockets.

Decimal system - Binary system

Since computers run on electricity, all of the internal components can only "understand" two states: they are either in a **low-voltage state** or in a **high-voltage state**. All modern computers are what we call **binary** machines. This means that the language they use internally to function is the **binary numeral system**, which is a way to write numbers using only two digits: **0** (low-voltage state) and **1** (high-voltage state).

By using a series of 0s and 1s, we can create all the other numbers. In the decimal number system, each digit can take one of ten values (0-9). When digits are put together to form a number, the position of each digit has a different place value, increasing by a power of ten.

To represent the number 131 in the decimal system:				
Digits	1	3	1	
Place value	10^2 =100	10^1 =10	10^0 =1	
	1×100 +	3×10 +	1×1 =	131



The same principle is used in the **binary system**. The difference is that each digit can take one of two values (0, 1), and each place value increases by a power of two (ones, twos, fours, eights, etc.).

Smart Tip

In computers, the basic unit of information is called a **bit**, and it can either be 0 or 1. The word comes from a contraction of "binary digit."

Now let's try to make a fill-in form. The steps you must follow are:									
Digits	1	0	0	0	0	0	1	1	
Place value	2^7 =128	2^6 =64	2^5 =32	2^4 =16	2^3 =8	2^2 =4	2^1 =2	2^0 =1	
	1×128 +	0×64 +	0×32 +	0×16 +	0×8 +	0×4 +	1×2 +	1×1 =	131

Notice that the place value of the rightmost digit in either system is **1**. Any number (except zero) raised to the zero power equals one. Consequently, $10^0=2^0=1$. You can now read and understand any number in the binary system!

0 1 0 0 1

2^4 2^3 2^2 2^1 2^0

Hexadecimal system

As computers got more powerful, they were able to work with more and more bits of information. The **hexadecimal numeral system** was employed by people working with computers in an attempt to shorten the very long binary series. The hexadecimal system is a base-16 number system, meaning that each digit can take any of 16 distinct values. Of course, in this case, we need symbols to represent the values 10, 11, 12, 13, 14, and 15. So, we use the letter A to represent the number 10, B to represent 11, C for 12, etc.

The 16 digits in base-16 are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

In the hexadecimal system, the place value increases by a power of sixteen (ones, sixteens, two hundred and fifty-sixes, four thousand and ninety-sixes, etc.).

So, let's convert the hexadecimal number 1C8A to decimal:					
Digits	1	C	8	A	
Place value	16^3 =4096	16^2 =256	16^1 =16	16^0 =1	
	1×4096 (=4096) +	12×256 (=3072) +	8×16 (=128) +	10×1 (=10) =	7306

Conversions between systems

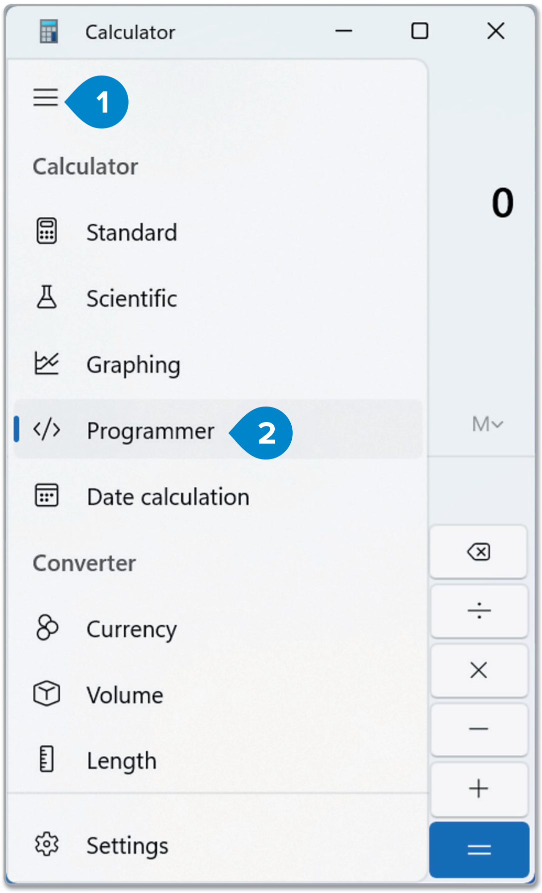
It is very easy to convert numbers from one system to another using Windows Calculator.

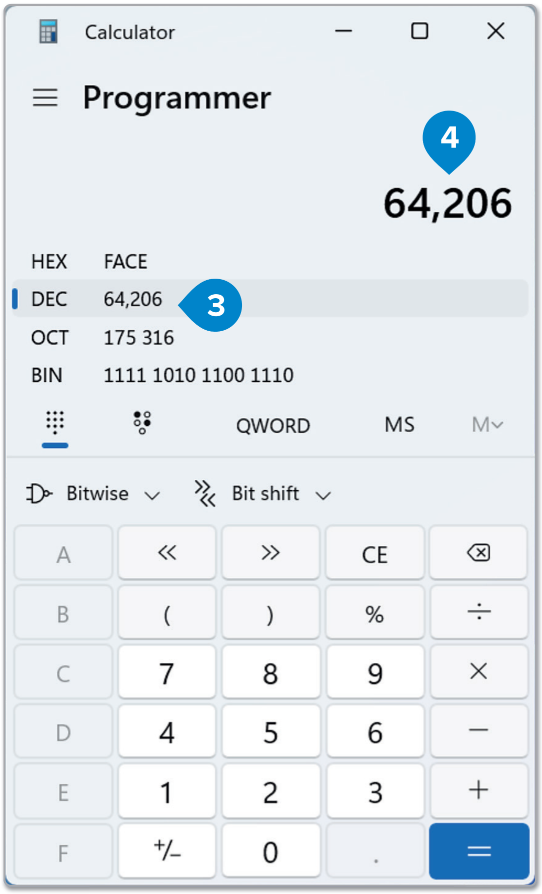
Since the hexadecimal system contains some letters, computer programmers like to create "magic numbers" that spell words and use them in their programs.

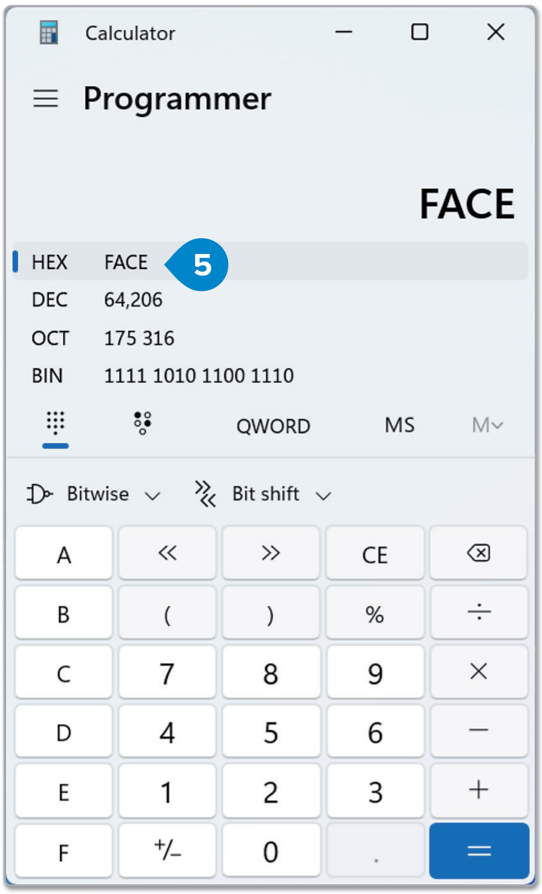
For example, DEADBEEF ("dead beef") is often used to indicate a software crash, and 8BADF00D ("ate bad food") is used by Apple in iOS when an application crashes.

To convert a number to a hexadecimal number:

> Open **Calculator**.
> Click **Open Navigation** 1 and click **Programmer** mode. 2
> Click a number system (e.g., DEC). 3
> Type a number into that number system. 4
> Click another number system to convert your number to. 5







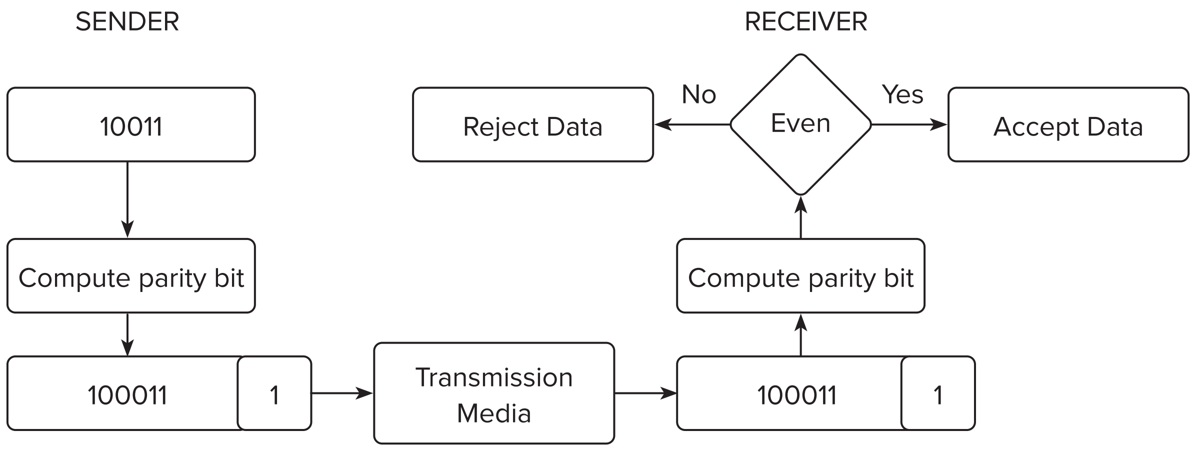
Error detection and correction

When data is transmitted or stored, errors can occur. These errors may result in corrupted information, compromising the reliability of the data. To ensure the accuracy of data, computers use techniques to detect and, in some cases, correct errors.

Error detection

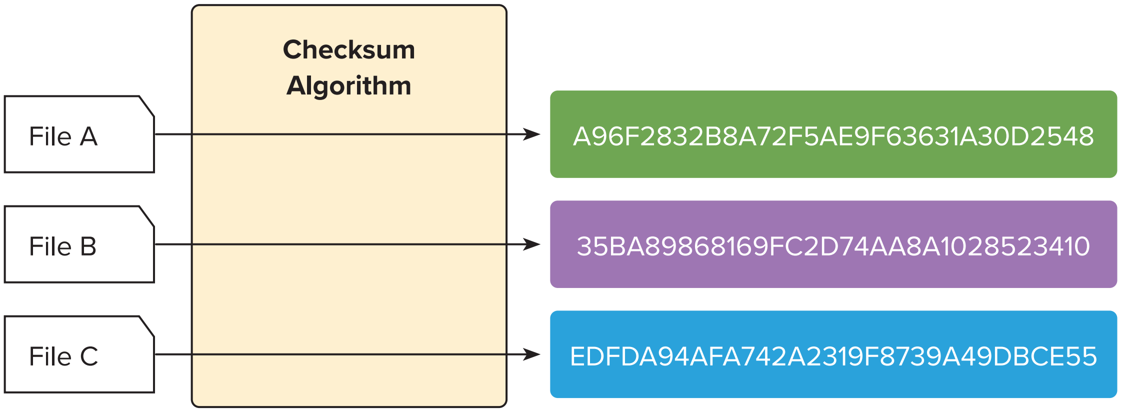
Errors can occur due to factors such as noise, interference, or hardware malfunctions. To ensure **data integrity**, computers use various methods to detect these errors. One common method is the **parity bit**, which is an extra bit added to a set of data bits. The purpose of this bit is to ensure that the total number of 1s in the data is either even (even parity) or odd (odd parity). For example, in even parity, if the number of 1s in a message is odd, the parity bit is set to 1 to make the total even. When the data is received, the system checks if the parity matches the expected value. If it doesn't, the system identifies that an error occurred during transmission. While this method can detect errors, it cannot pinpoint or fix them.

The following diagram illustrates the process of an even parity bit error detection:



Another widely used error detection technique is the **checksum**. A checksum is a numerical value calculated based on the contents of the data being transmitted. The sender calculates this value and sends it along with the data. When the data is received, the system recalculates the checksum and compares it to the original value. If they don't match, it indicates that the data has been corrupted, although like parity bits, checksums are also limited to detecting errors without correcting them.

This demonstrates the checksum error detection technique:



Error correction

While detecting errors is important, some systems also need to correct them, especially when data transmission is critical. **Hamming code** is a common method used for both detecting and correcting errors. This technique works by adding multiple check bits to the data at strategic positions. These bits help identify not only if an error occurred but also the location of the error within the data. When the system detects an error, it can automatically correct a single-bit error by flipping the incorrect bit to its correct value.

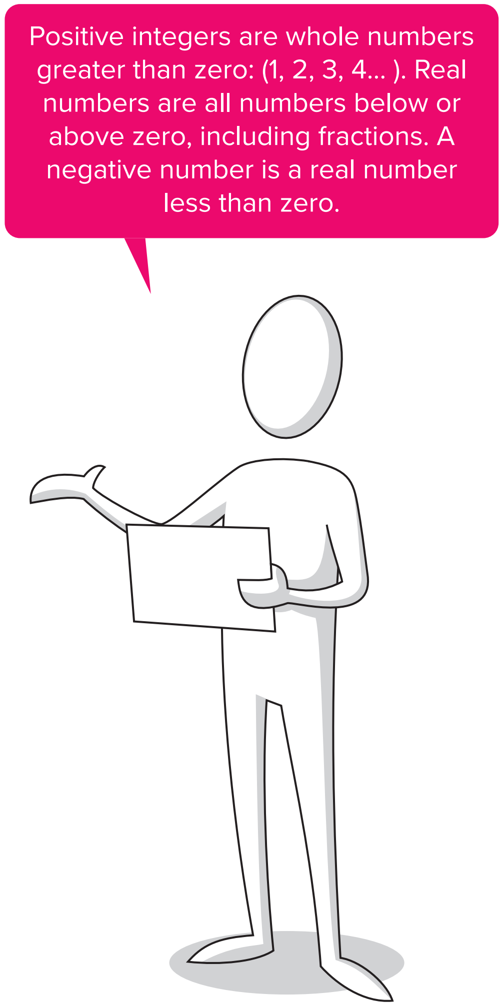
In more advanced systems, techniques such as **Reed-Solomon codes** are used to correct multiple errors within larger data sets. This is often used in technologies like CDs and DVDs, where data must be read accurately despite potential damage to the physical media. These advanced error correction techniques ensure data integrity by enabling the system to reconstruct the original data, even when multiple errors are present.

Data representations

Every computer works by manipulating data in various ways. Because computers can only store and manipulate data using binary digits, we need ways to represent many different types of data like numbers, text, images, and video in binary.

As for numbers, we have already discussed how we can represent decimal positive integers using the binary system. To work with negative or real numbers, there are various representation systems, which help us use any number through a string of 0s and 1s.

To represent **text** in computers, we use character sets. A **character set** is simply a list of characters and the binary code used to represent each one. One of the most used character sets is the **ASCII character set**. ASCII stands for American Standard Code for Information Interchange.



History

The Hamming code was invented by Richard W. Hamming, an American mathematician and computer scientist, in 1950. The Hamming code was one of the first error-correcting codes and laid the foundation for modern data communication and storage systems.

The first 32 characters in the ASCII character chart do not have a simple character representation that you can print to the screen. These characters are reserved for special purposes such as the Enter and Tab characters in a text file.

The codes in this chart are expressed as decimal numbers, but these values get translated to their binary equivalent for storage in the computer.

ASCII character reference table with hexadecimal and decimal values

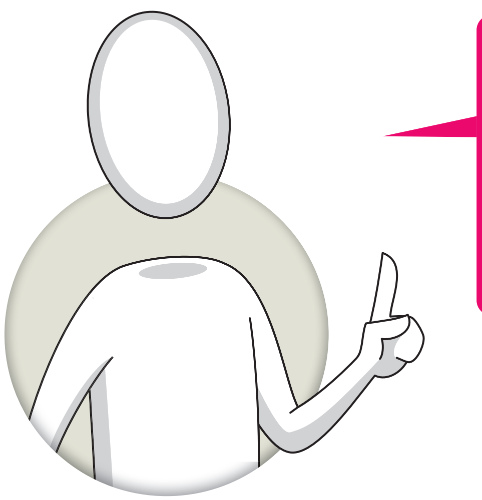
Dec	Hex	Char	Action (if non-printing)	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	NUL	(null)	32	20	Space	64	40	@	96	60	`
1	1	SOH	(start of heading)	33	21	!	65	41	A	97	61	a
2	2	STX	(start of text)	34	22	"	66	42	B	98	62	b
3	3	ETX	(end of text)	35	23	#	67	43	C	99	63	c
4	4	EOT	(end of transmission)	36	24	$	68	44	D	100	64	d
5	5	ENQ	(enquiry)	37	25	%	69	45	E	101	65	e
6	6	ACK	(acknowledge)	38	26	&	70	46	F	102	66	f
7	7	BEL	(bell)	39	27	'	71	47	G	103	67	g
8	8	BS	(backspace)	40	28	(	72	48	H	104	68	h
9	9	HT	(horizontal tab)	41	29	)	73	49	I	105	69	i
10	A	LF	(NL linefeed, new line)	42	2A	*	74	4A	J	106	6A	j
11	B	VT	(vertical tab)	43	2B	+	75	4B	K	107	6B	k
12	C	FF	(NP linefeed, new page)	44	2C	,	76	4C	L	108	6C	l
13	D	CR	(carriage return)	45	2D	-	77	4D	M	109	6D	m
14	E	SO	(shift out)	46	2E	.	78	4E	N	110	6E	n
15	F	SI	(shift in)	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	(data link escape)	48	30	0	80	50	P	112	70	p
17	11	DC1	(device control 1)	49	31	1	81	51	Q	113	71	q
18	12	DC2	(device control 2)	50	32	2	82	52	R	114	72	r
19	13	DC3	(device control 3)	51	33	3	83	53	S	115	73	s
20	14	DC4	(device control 4)	52	34	4	84	54	T	116	74	t
21	15	NAK	(negative acknowledge)	53	35	5	85	55	U	117	75	u
22	16	SYN	(synchronous idle)	54	36	6	86	56	V	118	76	v
23	17	ETB	(end of trans. block)	55	37	7	87	57	W	119	77	w
24	18	CAN	(cancel)	56	38	8	88	58	X	120	78	x
25	19	EM	(end of medium)	57	39	9	89	59	Y	121	79	y
26	1A	SUB	(substitute)	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	(escape)	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	(file separator)	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	(group separator)	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	(record separator)	62	3E	>	94	5E	^	126	7E	~
31	1F	US	(unit separator)	63	3F	?	95	5F	_	127	7F	DEL

To store **images**, we need to represent the color of each pixel in an image. The most common way is to use the **RGB model** in which each color is the sum of the different shades of the three primary colors (Red, Green, and Blue). So, for each pixel, we actually store three values, one for each color, each ranging from 0 to 255, which indicates the shade of each color from black to pure red, for example. Thus, an image is the binary representation of three colors that make up the pixels of the picture.

Monitor color analysis

Monitor color	R	G	B
White	255	255	255
Red	255	0	0
Green	0	255	0
Blue	0	0	255
Cyan	0	255	255
Magenta	255	0	255
Yellow	255	255	0
Black	0	0	0

Video is the most complex data type to represent, but generally it can be thought of as a series of images, saved in binary and played back one after the other. These images are usually compressed to save storage space and process the images as fast as possible.



Besides RGB, other color models are used in digital media for different purposes. The CMYK (Cyan, Magenta, Yellow, Key/Black) color model is mainly used in printing because it subtracts light from white paper.

Hands on!

1. **Read the following sentences and put a check mark for True or False.**

	True	False
1. The digit "0" represents the high-voltage state.		
2. In a hexadecimal system, each digit can take any of 16 distinct values.		
3. The binary system uses ten digits (0-9) to represent numbers.		
4. A checksum can detect and correct errors in the transmitted data.		
5. A parity bit is an extra bit added to a set of data bits.		
6. The parity bit can ensure data integrity by detecting errors.		
7. Reed-Solomon codes are used for correcting multiple errors in data sets.		
8. The RGB model is used to represent the color of each pixel in an image.		
9. The color yellow in the RGB model is represented by the values 255, 255, 0.		

1. **Explain how a parity bit works in error detection.**

2. Given that images and text are represented using binary data, why is it important to have error correction techniques in place when transmitting large files like images or documents over a network?

3. Convert each number from one system to the others and fill in the table.

Binary	Decimal	Hexadecimal
1101		
	85	
		3F8

4. Represent the characters in the decimal and binary system, based on the ASCII character set.

Character	Decimal	Binary
M		
n		
@		

LESSON 2

Logic circuits

How do computers perform operations using only binary digits?
How do computers process input signals into outputs?

Boolean logic

The data is in binary format, but computers have to convert the data for it to do something useful.

Boolean logic, named after its creator, mathematician George Boole, provides us with a set of operators that do simple transformations and comparisons of data. Just like in decimal arithmetic that humans use, where we have operations like addition, subtraction, multiplication, and division, in Boolean logic, we have **NOT**, **AND**, **OR,** and **XOR**. With these basic operators, we can create complex systems, including something as sophisticated as a computer CPU!

Logic gates

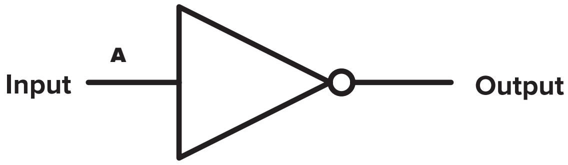
In computers, each basic Boolean operation is carried out by a **logic gate**, a device that receives one or more input signals and generates a single output signal.

Each gate is associated with what is called a **truth table**. A truth table lists all possible input values and the corresponding output values for a specific gate.

Logic Gate NOT

A **NOT** gate accepts one input value and produces one output value. The NOT operator inverts the input. If the input is 0, the output is 1 and if it is 1, the output is 0.

Input	Output
A	NOT A
0	1
1	0

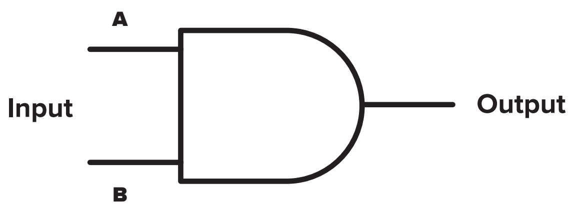


Smart Tip
Did you know that you can use Boolean operators to refine your web searches? Using the NOT operator (-) you can exclude results that contain a particular word. For example, searching [jaguar speed -car] will give you the speed of the animal excluding references to the car with the same name. Also, by default every space in your search query is considered to be an AND operator, meaning that you want the results to contain all the words of your query.

Logic Gate AND

An **AND** gate accepts two input values, which both determine the output. If the two input values are both 1, the output is 1; otherwise, the output is 0.

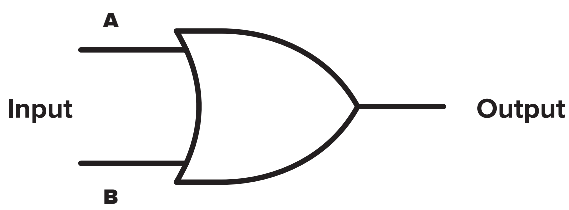
Input		Output
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1



Logic Gate OR

Like the AND gate, an **OR** gate has two inputs. If the two input values are both 0, the output value is 0; otherwise, the output is 1.

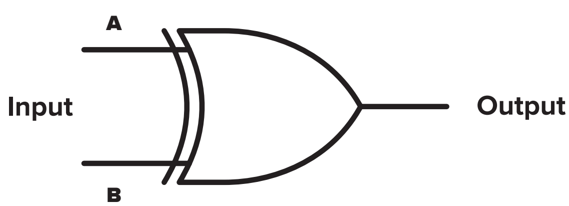
Input		Output
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1



Logic Gate XOR

An **XOR**, or exclusive OR, gate produces 0 if both inputs are the same, and 1 if they are different.

Input		Output
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



Combining logic gates

Logic gates can be combined to form more complex inputs and outputs. These combinations are known as **logic circuits**, which are the foundation of all digital systems. By connecting multiple logic gates in various ways, computers can perform complex operations such as arithmetic calculations, data storage, and decision-making processes.

Calculating the outputs of logic gates

Let's go step by step to understand how the outputs of the logic gates are calculated based on the following circuit diagram. By analyzing the circuit structure, it becomes easier to break down complex operations and complete the truth table accurately.

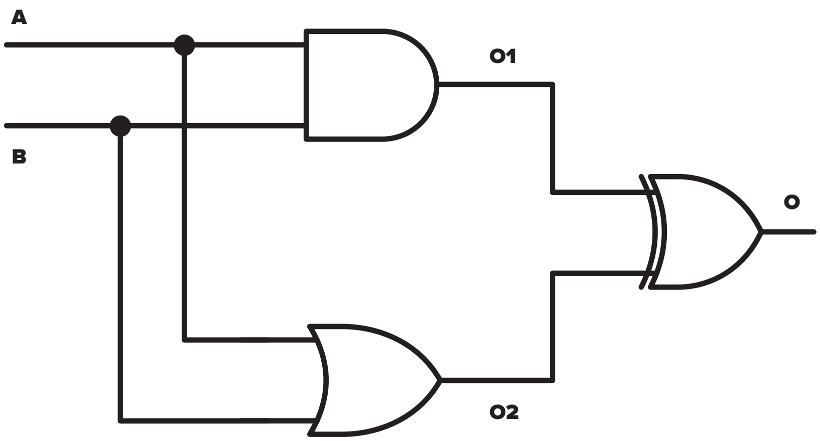
Step 1: Understand the circuit
Understanding the circuit is crucial for accurately identifying the logic gates and how they process inputs to produce outputs. This step ensures the correct flow of signals through the circuit, preventing calculation errors and leading to accurate results.

The circuit has:

- An **AND** gate producing O1 with inputs A and B.
- An **OR** gate producing O2 with inputs A and B.
- A final **XOR** Gate producing O with inputs O1 and O2.

The goal is to calculate the outputs (O1, O2, O) for all possible input combinations of A and B.

Input		Output		
A	B	O1	O2	O
0	0			
0	1			
1	0			
1	1			



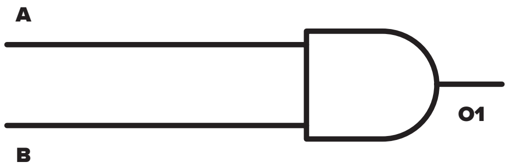
By carefully analyzing each input combination, errors are minimized, and the truth table can be completed correctly, reflecting the true function of the logic circuit.

Smart Tip
When solving circuits with multiple gates, filling in one column of the truth table at a time keeps your work organized and minimizes mistakes.

Step 2: Calculate O1 (AND Gate)
Analyzing each input combination is important because it allows for a step-by-step evaluation of how the circuit responds to all possible input values.

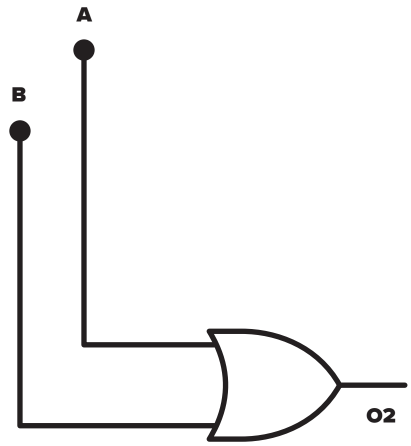
The AND Gate outputs 1 only if both inputs are 1. So based on the input vales, the truth table is:

Input		Output		
A	B	O1	O2	O
0	0	0		
0	1	0		
1	0	0		
1	1	1		



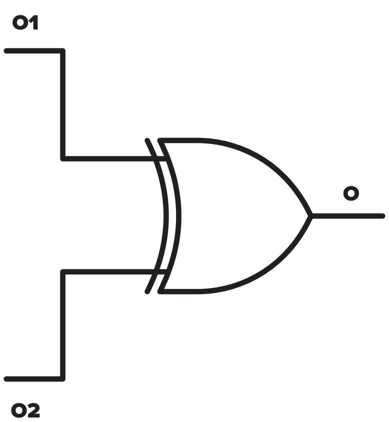
Step 3: Calculate O2 (OR Gate)
The OR Gate outputs 1 if at least one input is 1 . So based on the input vales, the truth table is:

Input		Output		
A	B	O1	O2	O
0	0	0	0	
0	1	0	1	
1	0	0	1	
1	1	1	1	



Step 4: Calculate O (XOR Gate)
The final XOR Gate takes **O1** and **O2** as inputs and produces 0 if both inputs are the same, and 1 if they are different.

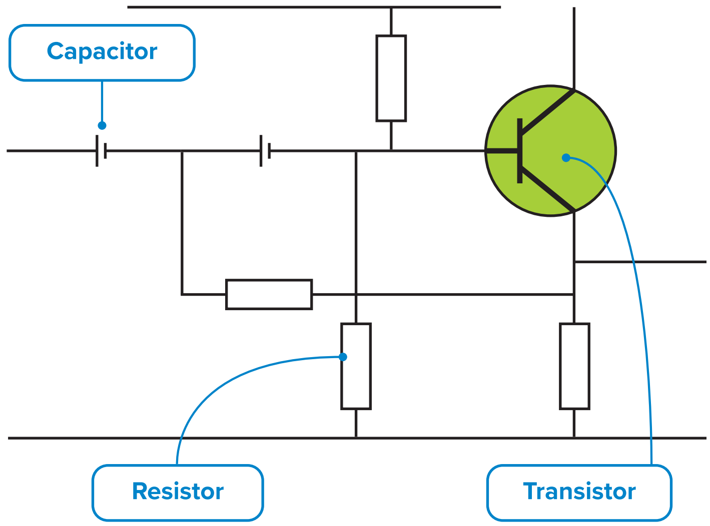
Input		Output		
A	B	O1	O2	O
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

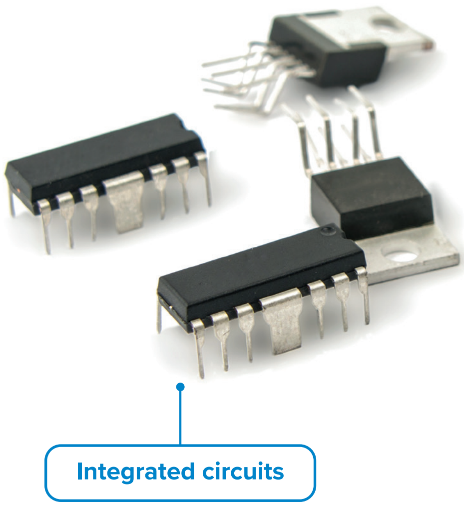


Transistors

Gates are made of one or more transistors. A **transistor** is a device that acts, depending on the voltage level of an input signal, either as a wire that conducts electricity or as a resistor that controls the flow of electricity.

Gates are used in all computer components, from the RAM modules to the little external flash memory sticks. These flash memory sticks use a special combination of gates that allows them to retain their state so that data can be stored permanently, with no need for electrical power to keep them saved after you remove the drive from your computer.

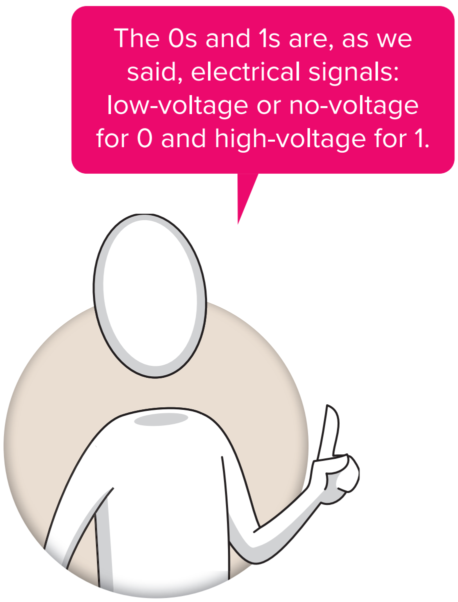




Integrated circuits

By combining multiple gates together and, in some cases, with other electrical elements like resistors and capacitors, we can create electronic circuits with multiple inputs and multiple outputs.

An **integrated circuit** (also called a chip or microchip) is a set of electronic circuits all enclosed in a small package. Microchips have revolutionized modern electronics design because of their very small size, low energy consumption, and very high potential to carry out complex tasks extremely fast.



Evolution of CPU transistor counts over the decades

Year	Processor Unit	Transistor Count	Designer
1971	CPU	2,300	Intel
1982	CPU	40,000	NEC
2012	CPU	5,000,000,000	Intel
2023	CPU	134,000,000,000	Apple
1997	GPU	3,500,000	NVIDIA
2012	GPU	7,080,000,000	NVIDIA
2024	GPU	104,000,000,000	NVIDIA

Hands on!

1. **Read the following sentences and put a check mark for True or False.**

	True	False
1. Microchips are large and heavy and consume a lot of energy.		
2. An integrated circuit is a set of electronic circuits all enclosed in a small package.		
3. Flash memory needs continuous electrical power to keep data saved.		
4. Gates in computers are made without using transistors.		
5. RAM modules and flash drives both rely on transistors to function.		
6. By 2012, Intel's CPU had reached 5 billion transistors.		
7. A transistor always allows electricity to flow, no matter the voltage.		
8. You can use Boolean operators to refine your web searches.		
9. By connecting multiple logic gates, computers can perform complex operations.		

2. Read the questions and put a check mark for the correct answer.

1. For which logic gate is the output 1 if the two input values are both 1?
   - a. XOR
   - b. NOT
   - c. AND
   - d. OR

2. What is the output of a NOT gate when the input is 1?
   - a. 0
   - b. 1
   - c. Depends on the gate
   - d. None of the above

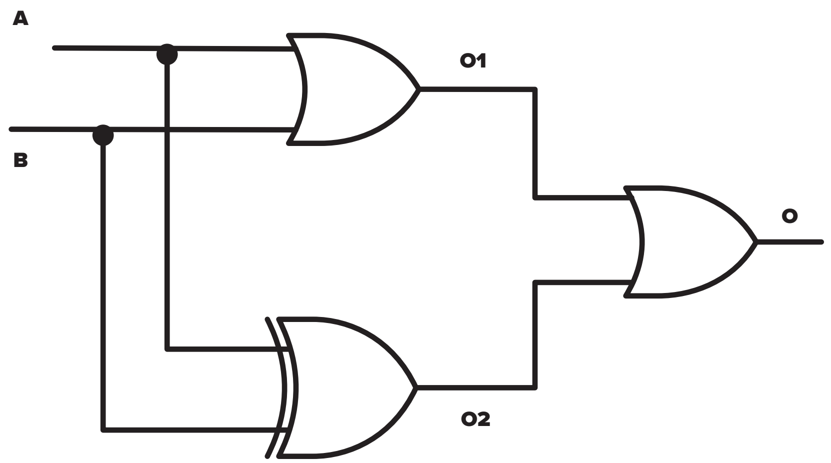
3. Which logic gate produces a 1 only if both inputs are different?
   - a. AND
   - b. OR
   - c. XOR
   - d. NOT

3. Why do you think the invention of integrated circuits was revolutionary for the design of modern electronic devices, particularly in terms of size, speed, and energy consumption?

4. Why do you think transistors are considered one of the most important inventions in modern technology? How would computers be different if transistors did not exist?

5. Find the outputs of the logic gates and fill in the truth table.

Input		Output		
A	B	O1	O2	O
0	0			
0	1			
1	0			
1	1			



LESSON 3

Operating systems

How does a computer display a website or play music when you ask it to? What separates a computer from a simple machine, allowing it to perform complex tasks?

Programs and software

In the previous tasks, we talked about the internal components and devices that make up a computer. All these parts, from the transistors and logic gates to the CPU and hard disk drives, form the **hardware** of the computer. However, a computer is not useful with hardware alone. It is now time to talk about **software**, which consists of all the instructions we provide to a computer for it to function and perform specific operations. A set of instructions designed for a particular task is called a **program**.

Program

A program is a specific set of instructions that a computer follows to perform a given task. For example, a web browser program contains instructions that tell the computer how to display web pages, load content, and handle user interactions, such as clicking links or entering text.

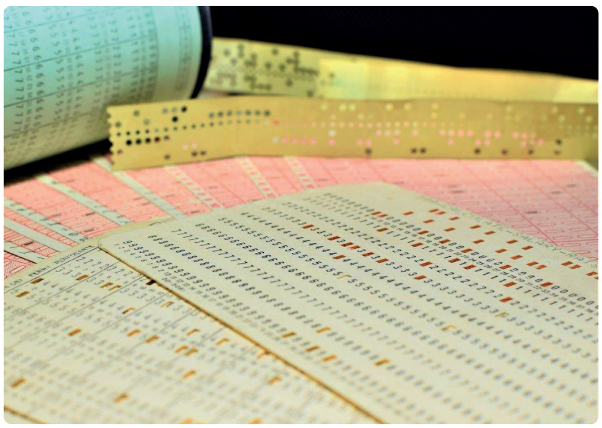
The operating system of a computer is the core of its system software. An operating system manages computer resources including memory and input/output devices, allows application software to access system resources, and provides a direct user interface to the computer system.

Software

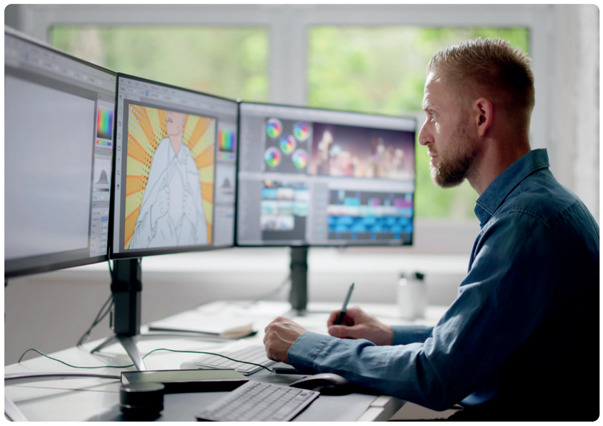
Software is a broader term that encompasses not just programs, but also the data and resources necessary for them to function. Software can include multiple programs working together, as well as libraries, utilities, and other components that enable the overall functionality of a system. An operating system is a type of system software that contains many different programs to help manage the hardware and provide an environment for application software.

The evolution of software

Software development has evolved significantly over time. In the early days of computing, software was often written for a specific hardware platform, and users had to input programs manually using punch cards or other methods. Early programs were very basic, with limited functionality compared to today's advanced software ecosystems.



As technology progressed, operating systems like **UNIX** and **Windows** were developed, providing a common platform for developers to create a vast spectrum of application software. Today, software development involves using sophisticated tools, such as **Integrated Development Environments (IDEs)** and version control systems to manage complex projects. As programming languages have evolved to become more efficient and easier to use, it allows developers to build increasingly complex applications.



Application software and System software

Modern software is divided into two categories: application software and system software.

Application software refers to programs designed to solve specific problems or assist users with tasks in everyday life. These programs run on top of system software and allow users to perform particular activities.

Examples of application software include:

- Word Processors (e.g., Microsoft Word, Google Docs)
- Web Browsers (e.g., Google Chrome, Mozilla Firefox)
- Games (e.g., Minecraft, Fortnite)
- Media Players (e.g., VLC Media Player, Windows Media Player)

Application software is typically more varied and user-facing compared to system software. Each program is usually specialized for one or a few tasks, and users can choose the programs that best fit their needs.

System software, on the other hand, is responsible for managing the computer system itself. It provides the tools and the platform in which application software can be created, run, and executed. System software interacts directly with the hardware and provides a layer of functionality that the hardware alone cannot provide.

System software includes software development tools, the programs that help us create application software etc.

Examples of system software include:

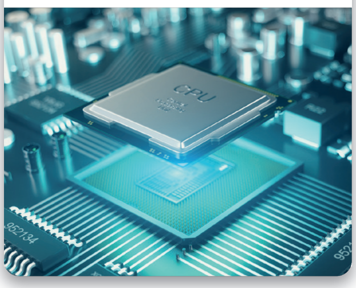
- **Operating Systems (OS)**: The most important piece of system software, the OS manages computer resources such as memory, input/output devices, and file systems while allowing application software to access system resources. Operating systems include Windows, macOS, Linux, and mobile operating systems like Android and iOS.
- **Utilities**: These provide additional tools to maintain and manage the computer system, such as antivirus programs, disk cleanup utilities, and file management software.
- **Device Drivers**: These programs allow the operating system to interact with and control hardware devices such as printers, video cards, and external drives. Without device drivers, the operating system would not be able to interface with peripherals or external devices.

How Application software and System software work together

Application software relies on system software to function correctly. Consider the following example:

- When you open a web browser, the operating system (system software) allocates memory and CPU resources to the browser so it can function. The browser itself is an application software that helps you interact with the web.
- If you were to print a document, the device driver (system software) for the printer allows the browser to send the document to the printer, which is an external hardware device.
- Utility software may scan your computer in the background, ensuring that your system remains free from malware or performance issues while you browse the web.

System software is like the foundation and operating environment that enables application software to run. Both types of software must work together seamlessly for a computer system to function effectively.

Hardware	↔	System software	↔	Application software
CPU Disk Printer Mouse		Operating system Utilities Device Drivers		Internet browsers Databases Word processors Spreadsheets Computer games
		Windows		W

Smart Tip

We can say that system software includes software development tools, the programs that help us create application software and other system software.

Multiprogramming

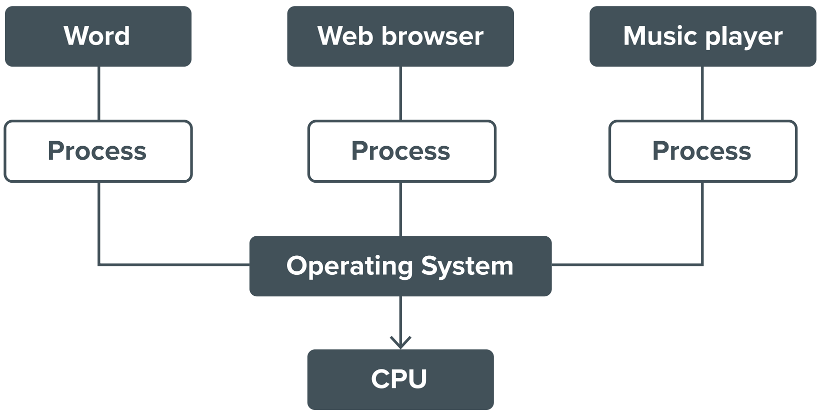
Multiprogramming is a fundamental feature of modern computer systems that allows multiple programs to reside in the computer's memory at the same time. Instead of executing one program entirely before moving on to the next, multiprogramming enables the operating system to manage multiple programs, allowing them to share CPU time. This technique helps the CPU make efficient use of its resources by reducing idle time.

In multiprogramming, several programs are loaded into memory, and the CPU rapidly switches between them, executing parts of each program. This gives the illusion that the programs are running simultaneously, even though the CPU can only execute instructions from one program at a time.

Imagine you're sitting at your desk, working on several tasks at the same time. You have:

- A word processor open where you are typing your school assignment.
- A music player running in the background, playing music.
- A web browser downloading a video for your class project.

Each of these tasks seems to be happening at the same time on your computer, but in reality, your computer's CPU can only focus on one task at a time. However, thanks to multiprogramming, your operating system manages these tasks in a way that makes them appear to run simultaneously.



How multiprogramming works

Let's examine the previous example to understand how multiprogramming works.

Task 1: Writing your assignment (Word processor)

You start typing on your word processor, and the computer dedicates some time to this task. The CPU receives and processes the instructions from the word processor, such as recording the letters you type, updating the display, and saving your document automatically.

Task 2: Playing music (Media player)

While you're typing, your music continues playing in the background. The CPU quickly switches to the media player to process its instructions, such as reading the audio file, sending the sound to your speakers, and controlling the play/pause buttons.

Task 3: Downloading a video (Web browser)

In the background, your web browser is downloading a video. Even though you're not actively interacting with it, the operating system ensures that the CPU periodically switches to the browser to download small chunks of the video file and store them on your computer.

Even though the CPU can only focus on one task at a time, it switches between these tasks so quickly—within milliseconds—that it seems like everything is happening at the same time. This rapid switching is called time-sharing.

Time slicing

In the above scenario, the CPU allocates a short period of time, known as a **time slice**, to each task. The CPU gives a time slice to the word processor so you can type a few words, then switches to the media player to ensure the music keeps playing and then to the web browser to download part of the video.

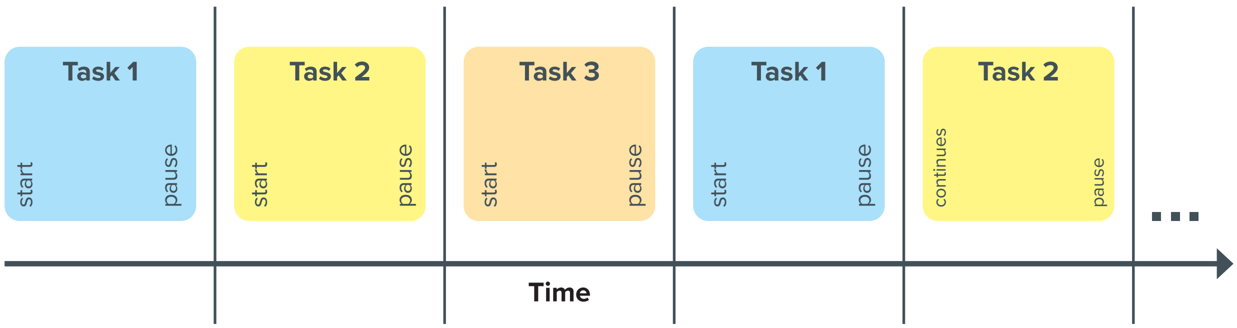
The operating system is responsible for deciding how long each task's time slice should be and when to switch between them. Once the time slice for one task is up, the CPU pauses that task and moves to the next one. After completing a time slice for the media player, for instance, it will return to the word processor, allowing you to keep typing without you even noticing the switch.

Process management

The operating system also manages the state of each task, known as a process. Each process can be in one of the following states:

- **Running**: The process currently has the CPU's attention.
- **Waiting**: The process is waiting for its turn to use the CPU.
- **Ready**: The process is ready to be executed as soon as the CPU becomes available.

When you're not actively interacting with the web browser, it might stay in the waiting state, but when it's time to download a new portion of the video, it will move to the running state for its time slice.



Hands on!

1. **Read the questions and put a check mark for the correct answer.**

   1. Which of the following is an example of system software?

      - [ ] a. Google Chrome
      - [ ] b. Microsoft Word
      - [ ] c. Windows Operating System
      - [ ] d. VLC Media Player

   2. Which statement correctly describes the difference between application software and system software?

      - [ ] a. System software is user-friendly, while application software is only for developers.
      - [ ] b. Application software manages hardware, while system software solves user problems.
      - [ ] c. System software manages the system and hardware, while application software helps users complete specific tasks.
      - [ ] d. Both application and system software serve the same function in a computer system.

   3. What is the main purpose of multiprogramming in modern computer systems?

      - [ ] a. To execute one program completely before starting another
      - [ ] b. To allow multiple programs to share CPU time, reducing idle time
      - [ ] c. To make the CPU faster by increasing its clock speed
      - [ ] d. To install more programs on the computer

   4. What is a time slice in computing?

      - [ ] a. A backup of all running programs
      - [ ] b. A short period of time allocated by the CPU to execute a task
      - [ ] c. A method of installing new software
      - [ ] d. A way to increase the CPU speed

2. Complete with the missing words.

1. A program is a specific set of ________________ that a computer follows to perform a given task.

2. As programming languages have evolved to become more ________________ and ________________ to use, it allows developers to ________________ increasingly complex applications.

3. ________________ is responsible for managing the computer system itself. It provides the ________________ and the platform in which application software can be created, run, and executed.

4. In multiprogramming, several programs are loaded into ________________, and the CPU rapidly ________________ ________________ them, executing parts of each program.

3. What is the difference between a program and software?

____

____

____

____

____

____

____

1. Foundations of computer science

4. List the main components of system software and briefly describe their functions.

5. In a multiprogramming system, how does the operating system make sure all programs get a fair amount of CPU time? What could happen if one program needs much more CPU time than others, and how does the operating system handle it?

LESSON 4

Abstractions

What is the primary purpose of abstraction in computing? Can you give an example of how abstraction is used in everyday life, outside of computing?

Abstractions in computing systems

Abstractions are tools that simplify complex systems by concealing the details of their implementation. In computing, abstractions make it easier for users to interact with technology without needing to understand how it works internally. For example, you use a smartphone to call someone or take a photo without thinking about the electronic components, algorithms, and processes that make these actions possible.

Embedded devices in everyday life

Many computing devices are integrated with other systems to improve functionality and efficiency. These are called **embedded systems**, and they can be found in various fields, including healthcare, transportation, and security.

Medical devices

Devices like pacemakers monitor and regulate health by performing specific functions autonomously. The user does not need to understand how the internal circuits or programming function—the abstraction allows the user to focus on its purpose rather than its complexity.

Hearing aids

These assistive devices amplify specific sound frequencies to help users hear better. The complex signal processing and frequency adjustments are hidden from the user, allowing easy control through simple buttons or a mobile app.

Vehicle monitors

Modern vehicles often include monitoring systems that track driving habits, fuel efficiency, or even provide navigation. The algorithms that process GPS signals or analyze driving patterns are abstracted away for the driver's convenience.

Facial recognition

Security systems often integrate facial recognition to identify individuals. The user interacts with a straightforward interface while the underlying system processes thousands of data points to match faces accurately.

Layers of a computing system

Computing systems are organized into different layers to reduce complexity. These **layers** work together, from the physical hardware to the application software you interact with. Understanding these layers helps explain how the commands you give to a device are processed and executed.

The three main layers

The hardware, system software, and application software layers work together to ensure computers run smoothly. These layers are interconnected—application software uses system software to work with the hardware, and system software controls the hardware to follow commands. This collaboration between layers helps computers operate quickly, reliably, and efficiently.

How the layers interact

Each layer communicates with the ones above and below it. For example:

1. When you press a key on your keyboard, the hardware detects the electrical signal.
2. The system software translates this signal into a binary value.
3. The application software interprets this binary value as a letter and displays it.

This layering makes it easier to design and use computing systems. Developers can focus on improving one layer without needing to redesign the others.

Abstractions across the layers

Abstractions exist at every layer of a computing system, simplifying interactions between them.

Hardware abstraction

The system software hides the complexity of the hardware from application developers. For example, you don't need to know how a CPU processes data to write a program.

System Software abstraction

Application developers rely on system software to handle tasks like memory management and device control, freeing them to focus on creating user-friendly applications.

Application abstraction

End-users interact with applications without needing to understand how the software processes their commands. For example, you can use a web browser without knowing how it communicates with servers.

A smart TV, for example, includes all three layers:

- **Hardware**: The screen, processor, and input devices (remote, buttons).
- **System Software**: The operating system that manages apps and settings.
- **Application Software**: Streaming apps like Netflix or YouTube.

The user interacts with the application software (selecting a show), while the underlying system software and hardware handle the rest.

Why abstractions matter

Abstractions reduce complexity, making computing systems easier to design, use, and maintain. Without abstractions, users would need to understand every detail of how their devices work, which would make technology less accessible.

Benefits of abstractions	
Feature	Description
Simplification	Allows users and developers to focus on their tasks rather than underlying complexities.
Efficiency	Speeds up development and usage by providing reusable frameworks and tools.
Accessibility	Makes technology usable for people with varying levels of technical expertise.

Abstractions and data representation

Another critical application of abstraction in computing is data representation. Computers rely on binary data to perform tasks, and abstraction helps us manage and interpret this data effectively.

As discussed in the previous lesson, computers use binary digits to represent various types of data. Here's how abstractions help:

1. **Numbers**: Beyond representing positive integers in binary, abstractions enable the representation of negative and real numbers through systems like two's complement and floating-point representation. These systems make it possible to process mathematical computations efficiently without manually handling binary logic.
2. **Text**: Character sets like ASCII provide an abstraction layer for text representation. Instead of remembering binary codes for each letter, programmers and users work with characters that the system automatically translates into their binary equivalents.
3. **Images**: The RGB model is another abstraction that simplifies how we store and manipulate visual data. Rather than working with raw binary values, the system uses predefined standards to represent colors, making image editing and display intuitive.
4. **Audio and Video**: Abstraction simplifies the handling of multimedia data. For instance, audio files use sampling rates and bit depth to encode sound in binary, while video files use codecs to compress and store sequences of frames efficiently.

Hands on!

1. Read the following sentences and put a check mark for True or False.

	True	False
1. Abstractions in computing help users interact with technology without needing to understand how it works internally.		
2. Embedded systems are only used in healthcare devices.		
3. The three main layers of a computing system are hardware, system software, and application software.		
4. When you press a key on your keyboard, the application software directly detects the electrical signal.		
5. Computing systems are divided into layers to make them easier to design and use.		
6. Audio files use sampling rates and bit depth to encode sound in binary form.		
7. Abstraction makes it easier to work with different types of data, such as text, images, audio, and video.		
8. The RGB model simplifies how we store and manipulate visual data.		
9. Vehicle monitors display all GPS signal processing steps to the driver for safety.		

2. Complete with the missing words.

1. ____________________ helps to reduce complexity in computing systems by hiding unnecessary details from the user.
2. The ____________________ acts as a bridge between hardware and application software, managing resources efficiently.
3. The ____________________ layer of a computing system includes applications like web browsers and games.
4. A computer's physical components, such as the CPU and memory, are part of the ____________________ layer.
5. A ____________________ set provides an abstraction layer for text representation.

3. Read the following descriptions and write the letter of the correct term.

A Abstraction

B System Software

C Embedded System

D Hardware Layer

1. A computing device integrated into another system for specific functions. ☐
2. Programs used by the end-user, like web browsers and games. ☐
3. The physical components of a computer, like the CPU and memory. ☐
4. A process that hides complex implementation details and displays only the essential features. ☐

4. Explain how abstractions simplify the interaction between different layers of a computing system.

5. Consider a modern smartphone. Identify three embedded systems within the device and explain how abstraction helps the user interact with them seamlessly without needing to understand the underlying technology.

LESSON 5

Network fundamentals

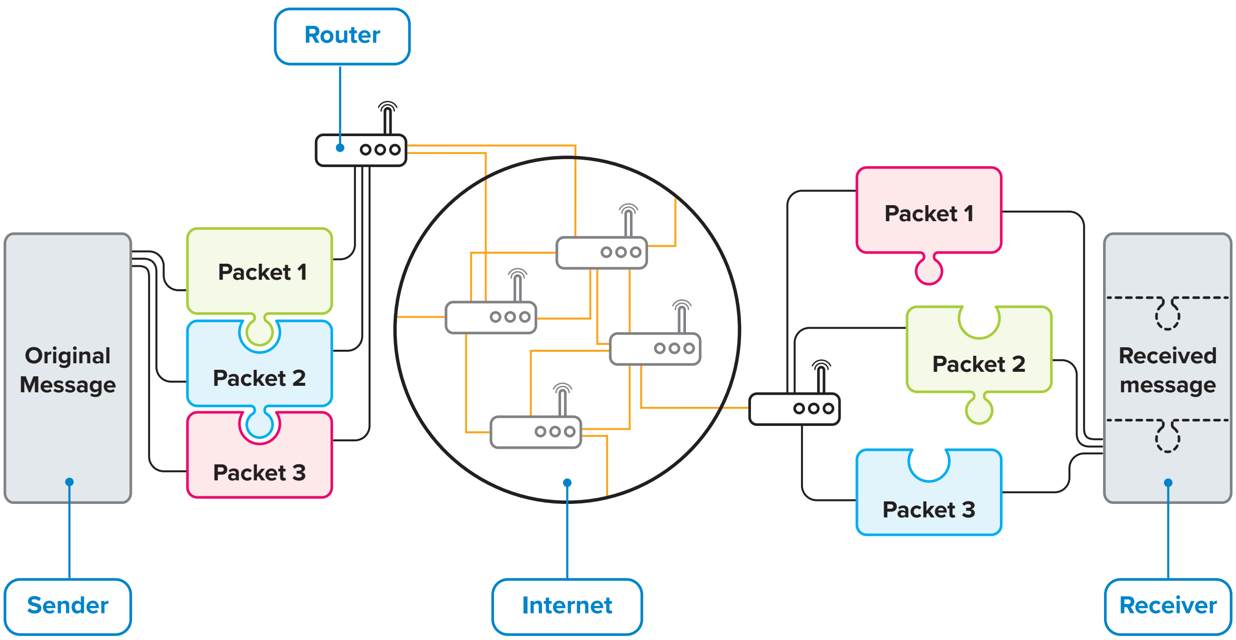
How is data sent without it getting lost? How does your computer block harmful websites while allowing safe ones?

Computers connect to each other and form networks in order to communicate and share resources. On a larger scale, every computer in the world can communicate with another through the Internet, the global network of networks. Now, it's time to explore some of the underlying mechanisms that enable information to travel through networks and reach its destination.

Packet switching

To achieve more efficient transfer of messages between different devices over networks, each message is divided into fixed-sized, numbered **packets**. The packets are then sent over the network to their destination, where they are received and reassembled to form the original message. This process is known as **packet switching**.

The individual packets of a message may take different routes in the network on their way to their destination. As a result, they may arrive in a different order than the one they were sent in. Once they arrive, the destination device must reorder them to recreate the original message.



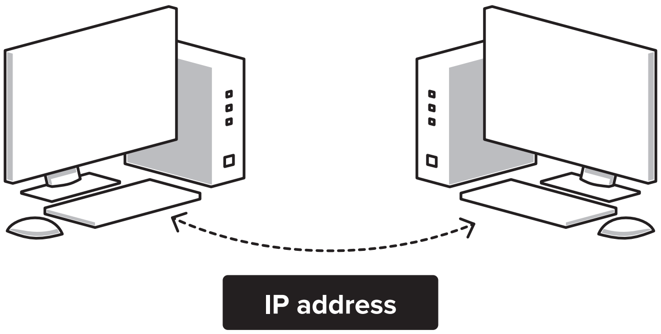
Routers

The reason different packets can reach the same destination through different paths lies in the design of all modern networks and, of course, the Internet as well. Traffic between networks is directed by network devices called **routers**. Neither your computer nor routers know the exact path a message from one computer to another must follow, especially inside a vast web of networks such as the Internet. What routers do know is the next best step toward a destination.

A router that receives a packet reads the destination and decides to which other router, directly connected to it, it must be forwarded to. The next router repeats this process until the packet reaches a router that is connected to the recipient device. If a certain path is blocked by a router malfunction, or if a path currently has a lot of network traffic, a router might send a packet along an alternative route. In this way, we achieve speed and reliability in transferring information over the network.

Network addresses

For two computers to communicate with each other, they need to identify each other from among all the other computers in the world. This is done in two ways:



A **hostname** is a unique name that specifies a particular computer on the Internet. Hostnames are generally readable words separated by dots.

For example: wikipedia.org

Although it is convenient for humans to use and remember hostnames, network devices like routers that transfer the actual messages back and forth use another kind of identification mechanism called an **IP address**. An IP address is a series of four decimal numbers separated by dots.

For example: **91.198.174.225**

Each of the four numbers that make up an IP address can be in the range 0–255.

As you can imagine, each hostname has a corresponding IP address. For us to be able to use convenient and easy-to-remember hostnames, we need a way to translate (resolve, as it is called) each hostname into the corresponding IP address. This is done automatically by the **Domain Name System** (**DNS**), which is a network of computers that constantly stores and provides the mappings from hostnames to IP addresses.

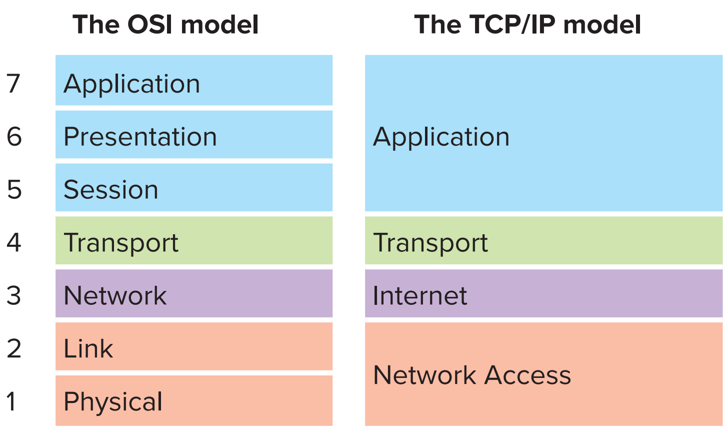
Protocols

A protocol is generally defined as a set of rules that governs how two entities communicate or interact. In networking, a **network protocol** refers to a set of rules that defines how data is formatted, transmitted, and processed across a network. Following is an illustration of the layers of the **Open Systems Interconnection (OSI)** model, which demonstrates how various networking protocols operate.

The OSI model

The development of a network application program is a fairly complex process. Starting from the transmission medium (cable) and ending at the final program, there are several stages in-between. In order to simplify this process, the ISO organization introduced the reference model OSI (Open Systems Interconnection).

The **OSI model** consists of seven layers. Each layer performs a specific task and serves the layer above it and is served by the layer below it. Changes happening in one layer do not affect the other layers.



OSI model layers		
Layer	**Name**	**Description**
7	Application	Interacts with software applications.
6	Presentation	Transforms data into the form that the application layer can accept, formats and encrypts data.
5	Session	Establishes, manages, and terminates the connections between the local and remote application.
4	Transport	Transfers data from a source to a destination host, with no errors in transmission.
3	Network	Transmits data from node to node. Uses routers and switches to control the flow.
2	Data Link	Packages raw bits from the physical layer into frames. Performs error detection.
1	Physical	Transmits raw bits through the physical medium.

TCP/IP

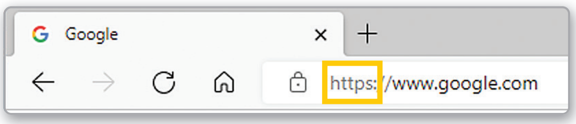
TCP stands for **Transmission Control Protocol** and **IP** stands for **Internet Protocol**. The name TCP/IP refers to a suite of protocols and utility programs that support low-level network communication. The name TCP/IP implies that TCP relies on the IP foundation below it. Those lower two layers of the **protocol stack** form the foundation of Internet communication.

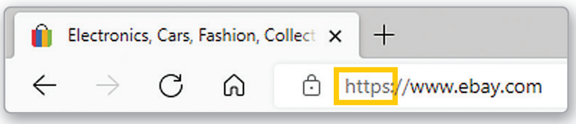
IP software is responsible for the routing of packets through the web of the various networks to their final destination. TCP software splits the messages into packets, passes them to the IP software for transmission, and then reorders and reassembles the packets at their destination. TCP software also deals with any errors that occur, such as if a packet never arrives at the destination or the contents of a packet are corrupted.

UDP stands for **User Datagram Protocol**. It is an alternative to TCP. The main difference is that TCP is highly reliable, at the cost of decreased performance, while UDP is less reliable, but generally faster. Note that UDP is part of the TCP/IP suite of protocols.

Essential high-level protocols above TCP/IP		
Acronym	Protocol name	Description
FTP	File Transfer Protocol	A protocol that allows a user on one computer to transfer files to and from another computer.
SMTP	Simple Mail Transfer Protocol	A protocol used to transfer electronic mail messages.
HTTP	Hypertext Transfer Protocol	A protocol defining the exchange of World Wide Web documents (web pages).
HTTPS	Hypertext Transfer Protocol Secure	A variant of HTTP offering secure communications between two computers.
DNS	Domain Name System	A system that translates hostnames of computers to corresponding IP addresses.

The HTTPS protocol is a powerful tool against online attacks. Essentially, it encrypts all traffic to and from a specific website, ensuring that when you access the site, no one can view your passwords or activities—even if a hacker intercepts the packets of your connection. Make it a habit to use HTTPS connections on sites that you have to log in to, such as Facebook and eBay, or even in search engines like Google to protect your search queries from others. All you have to do is add the "s" after "http" on your browser's address bar. It's also a good idea to update your bookmarks so you don't have to add the "s" manually every time.





The World Wide Web

One of the most common mistakes many people make when discussing networks is that they use the terms Internet and Web interchangeably.

The Internet is the global network that allows any computer connected to it to communicate with all others. The Internet is a medium through which information travels.

On the other hand, the World Wide Web, or simply the Web, is a specific way of accessing information over the medium of the Internet. The Web is a system of interlinked documents called web pages that are accessible via the Internet. The term interlinked means that each web page can link to one or more other web pages. The Web uses the HTTP protocol, one of the many "languages" spoken over the Internet, to transmit data.

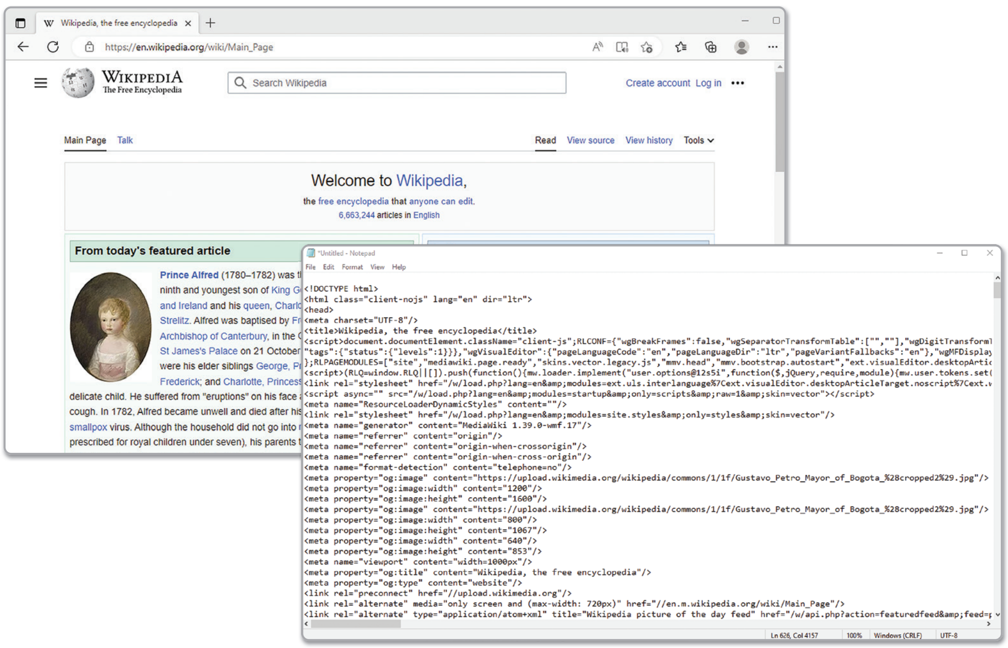
To access web pages, we use programs called web browsers, which allow us to read these documents and click links to navigate to other pages. These links are called **hyperlinks**.

Each page on the Web is uniquely identifiable by an address called a **URL**, which stands for **Uniform Resource Locator**.

For example: **https://en.wikipedia.org/wiki/World_Wide_Web**

Note that a URL contains a hostname (en.wikipedia.org) as well as other information used to locate a particular document offered by the specific host.

Web pages are created with Hypertext Markup Language, or HTML. Although the name hypertext suggests pages that contain text and links to other pages, today it is most common for a web page to contain or link to other types of media as well, such as images, audio, and video.



History

The TCP/IP protocol suite resulted from research and development conducted by DARPA, a US Department of Defence agency, in the early 1970s. It was first used in ARPANET, the world's first operational packet switching network and the "father" of what was to become the global Internet.

Firewalls

A **firewall** is a software or hardware-based network security system that controls the incoming and outgoing network traffic by analyzing the data packets and determining whether they should be allowed through or not. It can be found as a software program running on your computer, embedded in networking hardware devices like routers or as a standalone device. A firewall creates a security barrier that separates and protects a single computer or a local network from the Internet.
Let's explore the evolution of firewalls to get a better idea of the functionality and capabilities of today's firewalls.

First generation

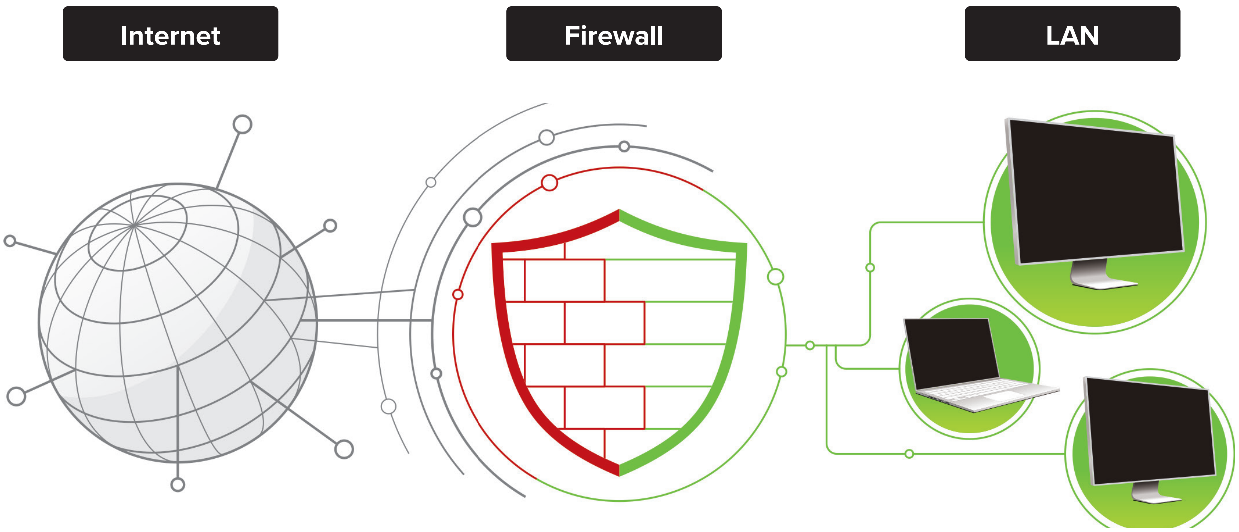
The first firewalls acted as **packet filters** by inspecting each individual packet that wanted to come in or out of the local network. The firewall filters packets based on the TCP/IP information they contain and decides if a packet will pass through by consulting a set of rules configured by the owner of the network under protection. For example, a firewall can be configured to only allow packets of a specific protocol and block all others or allow packets coming from a specific server.

Second generation

Second-generation firewalls perform the work of their first-generation predecessors but retain packets until enough information is available to make a judgment about its state. Known as **Stateful Packet Inspection**, it records all connections passing through it and determines whether a packet is the start of a new connection, a part of an existing connection, or not part of any connection.

Third generation

The latest firewalls, called **application layer firewalls**, are able to inspect traffic by filtering high-level protocols like FTP, DNS, and HTTP and block any unwanted protocols or detect if a protocol is being abused in any harmful way. They can "understand" the use of data packets and reject anything that is strange.



History

Sir Tim Berners-Lee is a British computer scientist, best known as the inventor of the World Wide Web. He implemented the first successful communication between two computers using the HTTP protocol in 1989.

Check your own firewall

Microsoft Windows comes equipped with a software firewall. Check whether it is on, and turn it on if it isn't.

To turn on the firewall:

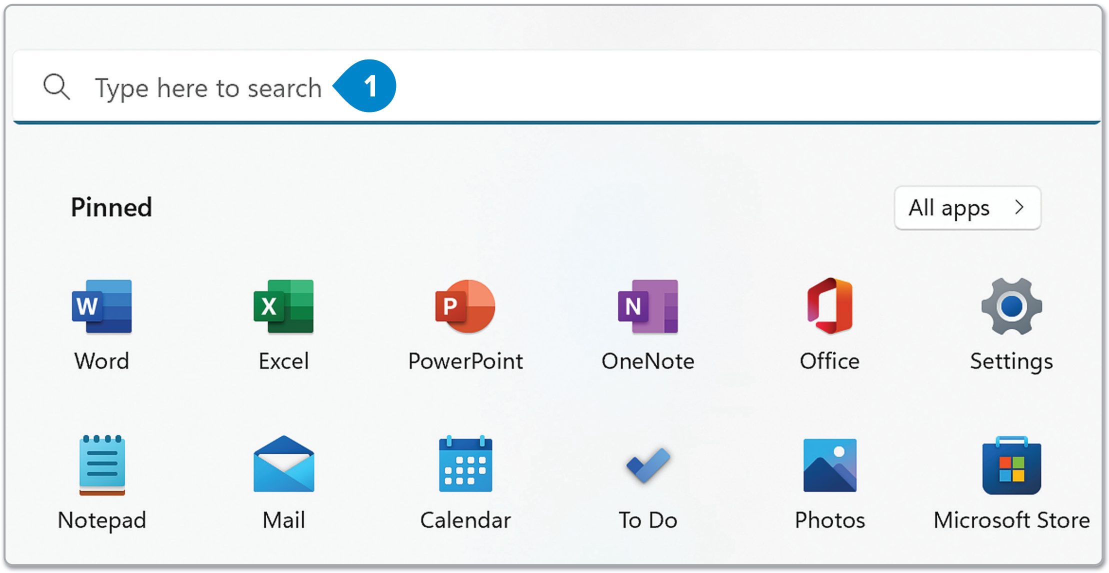
> Click the search box 1 and type "**control panel**". 2 Click the **Control Panel** application. 3

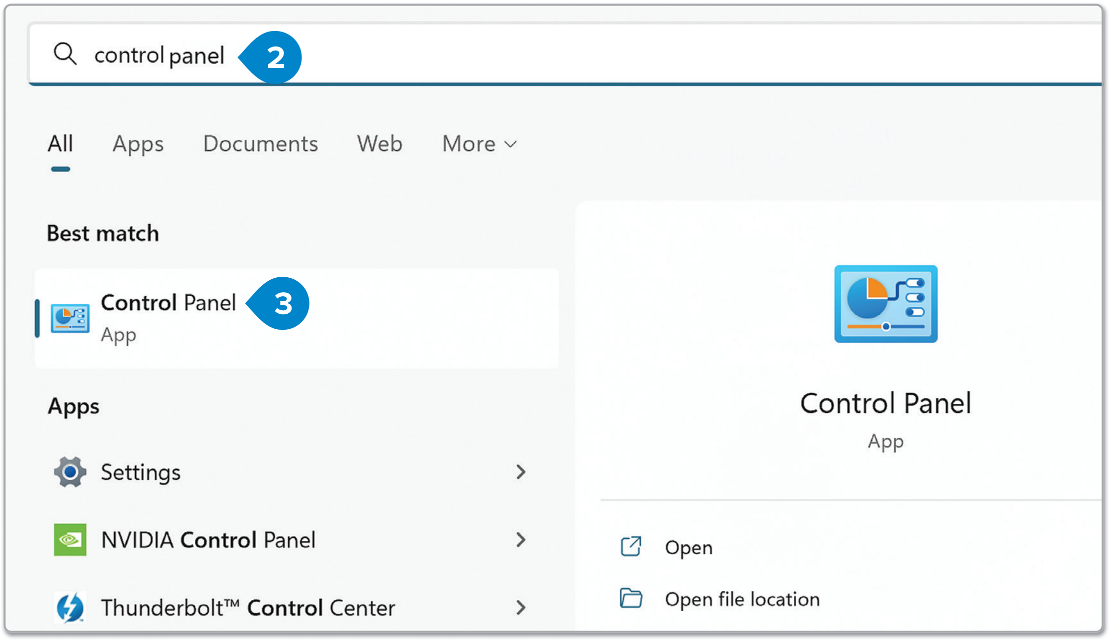
> In the search box, type "**firewall**" 4 and then click **Windows Defender Firewall**. 5

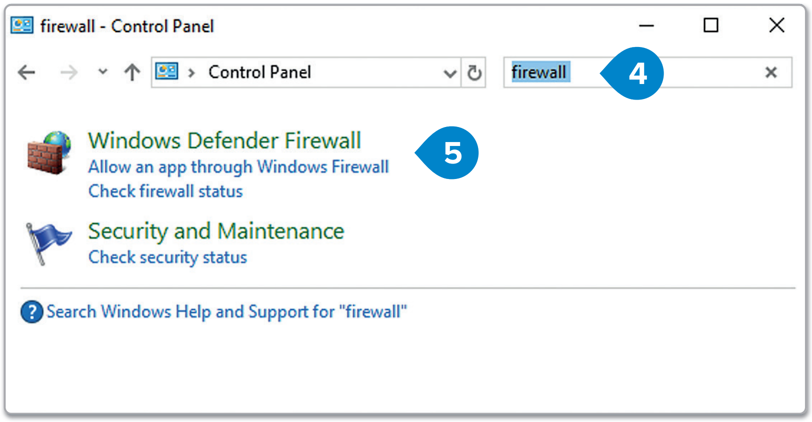
> If everything is green, then your firewall is enabled.

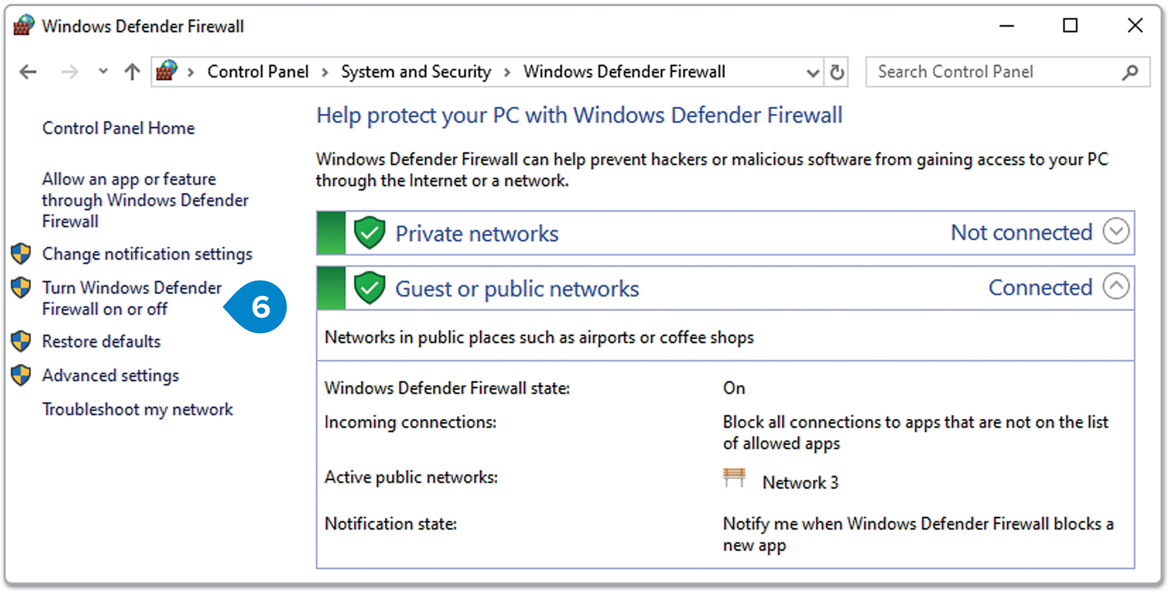
> If the firewall is off, click **Turn Windows Defender Firewall on or off**. 6

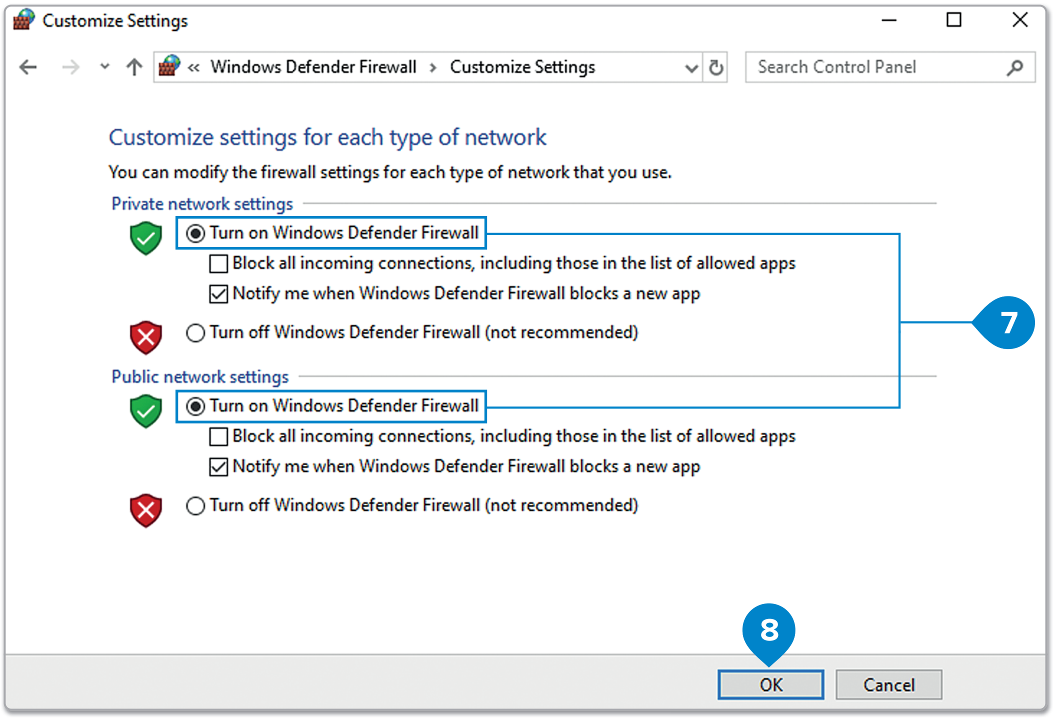
> Click **Turn on Windows Defender Firewall** 7 for all your networks and click **OK**. 8






firewall - Control Panel
Control Panel
firewall
4
Windows Defender Firewall
Allow an app through Windows Firewall
Check firewall status
5
Security and Maintenance
Check security status
Search Windows Help and Support for "firewall"


Windows Defender Firewall
Control Panel > System and Security > Windows Defender Firewall
Search Control Panel
Control Panel Home
Allow an app or feature through Windows Defender Firewall
Change notification settings
Turn Windows Defender Firewall on or off
6
Restore defaults
Advanced settings
Troubleshoot my network
Help protect your PC with Windows Defender Firewall
Windows Defender Firewall can help prevent hackers or malicious software from gaining access to your PC through the Internet or a network.
Private networks
Not connected
Guest or public networks
Connected
Networks in public places such as airports or coffee shops
Windows Defender Firewall state:
On
Incoming connections:
Block all connections to apps that are not on the list of allowed apps
Active public networks:
Network 3
Notification state:
Notify me when Windows Defender Firewall blocks a new app


Customize Settings
« Windows Defender Firewall > Customize Settings
Search Control Panel
Customize settings for each type of network
You can modify the firewall settings for each type of network that you use.
Private network settings
Turn on Windows Defender Firewall
Block all incoming connections, including those in the list of allowed apps
Notify me when Windows Defender Firewall blocks a new app
Turn off Windows Defender Firewall (not recommended)
7
Public network settings
Turn on Windows Defender Firewall
Block all incoming connections, including those in the list of allowed apps
Notify me when Windows Defender Firewall blocks a new app
Turn off Windows Defender Firewall (not recommended)
8
OK
Cancel

Hands on!

1. **Read the following sentences and put a check mark for True or False.**

	True	False
1. Packets are small, fixed-sized chunks of data transferred through a network.		
2. Packet Switching is the process by which packets are transferred all together through a network to their destination.		
3. Routers are software applications that carry forward packets between computer networks.		
4. When a packet arrives, the router reads its final destination and sends it directly and immediately.		
5. A hostname is a unique name that is assigned to a computer connected to the Internet.		
6. A URL is a character string that uniquely identifies a computer.		
7. An IP address is a series of four decimal numbers, which corresponds to four computers.		
8. A Domain Name System (DNS) is a network of computers.		
9. The OSI model consists of eight layers: Application, Presentation, Session, Transmission, Transport, Network, Link, Physical.		

2. Match the following acronyms of the protocols with their correct descriptions.

1. SMTP
2. HTTP
3. HTTPS
4. DNS
5. FTP

a. A protocol that allows a user on one computer to transfer files to and from another computer.

b. A protocol used to transfer electronic mail messages.

c. A protocol defining the exchange of World Wide Web documents (web pages).

d. A variant of HTTP offering secure communications between two computers.

e. A system that translates hostnames of computers to corresponding IP addresses.

3. Read the questions and put a check mark for the correct answer.

1. What is the primary function of a firewall in a network security system?

   a. To speed up Internet connections by filtering out slow websites

   b. To control incoming and outgoing network traffic by analyzing data packets and deciding whether to allow them through

   c. To store backup copies of important data

   d. To repair hardware components in networking devices

2. What is the key feature of second-generation firewalls?

   a. They only allow traffic from trusted websites.

   b. They act as packet filters using TCP/IP rules.

   c. They use Stateful Packet Inspection to track and analyze ongoing connections.

   d. They can block any device from accessing the Internet.

3. Which type of firewall can analyze high-level protocols like HTTP and detect harmful behavior?

   - [ ] a. First-generation firewall
   - [ ] b. Packet filtering firewall
   - [ ] c. Second-generation firewall
   - [ ] d. Application layer firewall

4. Explain what packet switching is, and describe how it ensures efficient and reliable data transfer across a network.

5. Explain the difference between TCP and UDP protocols.

6. What might be some limitations or challenges in relying solely on firewalls for network security?

Project

Computer evolution and history

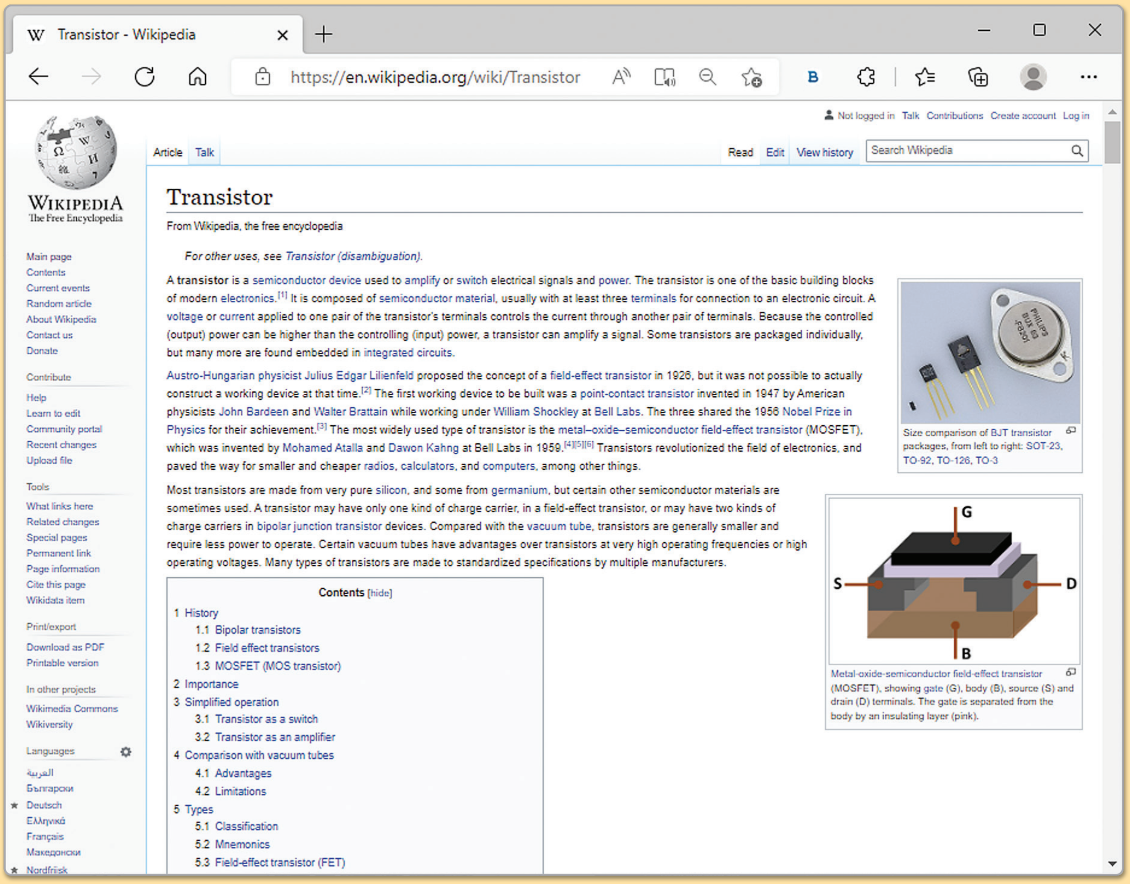
In this project, you are going to create a presentation about computers and their evolution through history. Begin by forming teams.

1. Try to answer the following questions:

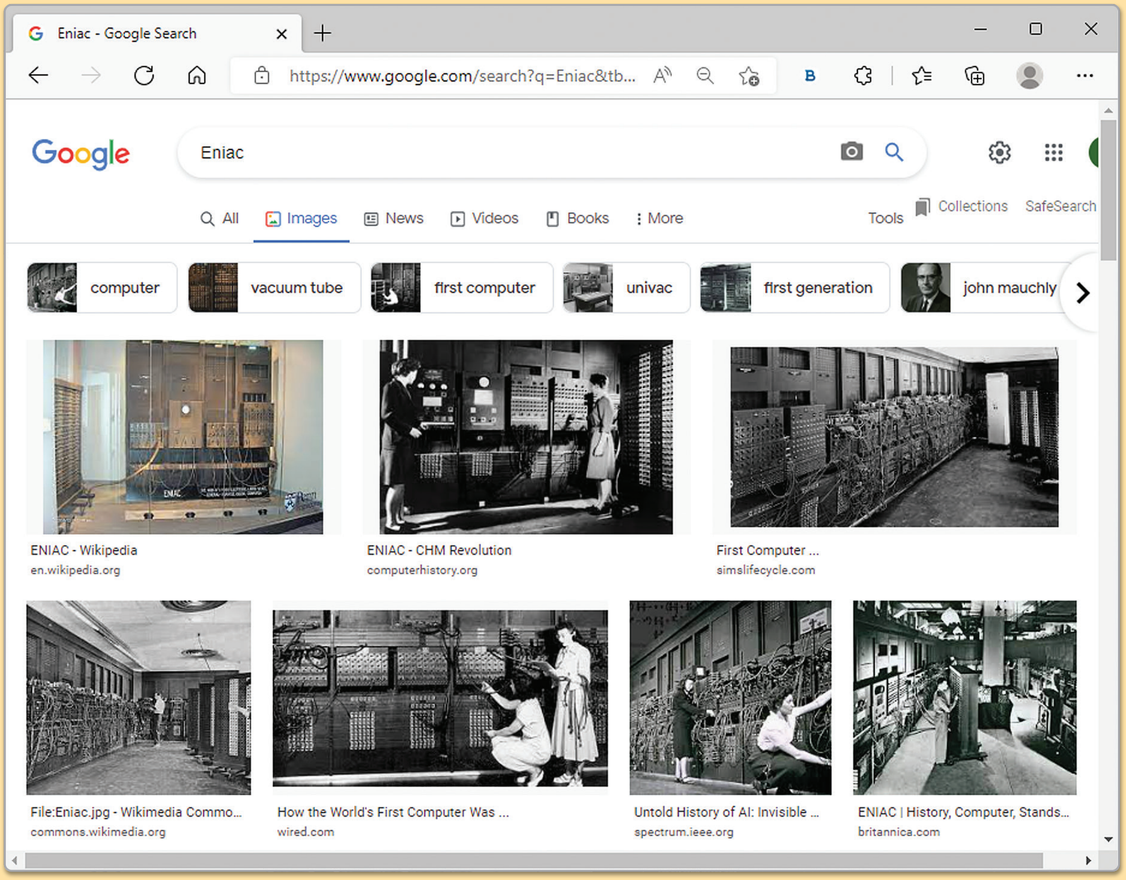
- What were the first computers like?
- What changes occurred from year to year as technology progressed?

2. Research the evolution of computer hardware. You will use the information you gather to create a timeline.

3. Search the Web for relevant information. Your goal should be to create a timeline of the most important milestones in the history of computer evolution.



4. Make your presentation more engaging with images of famous computers, computer scientists, and hardware components.



5. Be sure to include a section about the evolution of the various operating systems that were used in the computers of each era.

6. Devote a section of your presentation to the most important computer scientists and their contributions to the evolution of computer systems.

7. When your presentation is complete, share it with your classmates. Remember to apply the presentation tips you have learned to deliver a clear and engaging presentation.

Reflect

1. What surprised you the most about the early designs of computers?

2. How did hardware advancements influence the development of new operating systems?

3. Which computer scientist stood out to you, and why?

Wrap up

Take a moment to reflect on your progress.

How confident are you in your ability to apply the following skills?

> I can convert numbers between decimal and binary.
> I understand how computers process information using binary.
> I understand how abstraction is applied.
> I understand the basic Boolean operators (AND, OR, NOT, XOR).
> I can design simple logic circuits based on Boolean expressions.
> I can explain the difference between hardware and software.
> I can describe how multiprogramming allows computers to run multiple tasks.
> I know the seven layers of the OSI model and their functions.
> I can check Windows Defender Firewall.
> I can identify the different generations of firewalls.

Key Terms

abstraction
application software
ASCII character set
binary numeral system
Boolean logic
checksum
data integrity
decimal system
device drivers
Domain Name System (DNS)
embedded device
embedded
system
error correction
error detection
firewall
Hamming code
hardware
hexadecimal numeral system
hostname
hyperlink
integrated circuit
Internet Protocol (IP)
IP address
logic circuit
logic gates
multiprogramming
Open Systems Interconnection (OSI) model
operating system (OS)
packet
packet filter
packet switching
parity bit
program
Reed-Solomon code
RGB model
router
software
system software
time slice
transistor
Transmission Control Protocol (TCP)
truth table
Uniform Resource Locator (URL)
User Datagram Protocol (UDP)
utilities