

8.3 Layers of a Computer System Revision Guide:

Lesson 1 - Get in Gear

Learning objectives

- Recall that a general-purpose computing system is a device for executing programs
- Recall that a program is a sequence of instructions that specify operations that are to be performed on data
- Explain the difference between a general-purpose computing system and a purpose-built device

Key vocabulary

Computer, system, device, program, software, instructions

Key Concepts:

1. What is a General-Purpose Computer?

- A **general-purpose computer** is a device designed to execute a **program**. A program is a set of instructions that tell the computer what operations to perform.
- **General-purpose** means it can do many different tasks depending on the program it is running. Unlike a **purpose-built device**, which is designed to do one specific task (e.g., a calculator), a general-purpose computer can be used for almost anything.

2. What is a Program?

- A **program** is a sequence of instructions that tells a computer what to do. The instructions can process data in many different ways, depending on what the program is designed to do.
- **Software** is another word for programs. The programs you use every day on your computer or phone (like games, apps, or word processors) are examples of software.

3. The Role of Instructions:

- A **computer** follows **instructions** to perform tasks. Whether it's solving a mathematical problem, playing a game, or processing data, the computer performs these tasks by following instructions in a **program**.
-

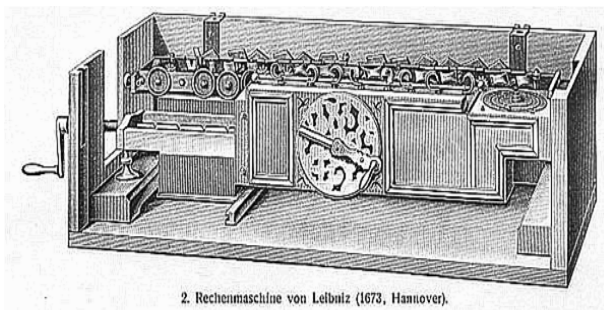
Comparing Early Calculating Machines with Modern Computers:



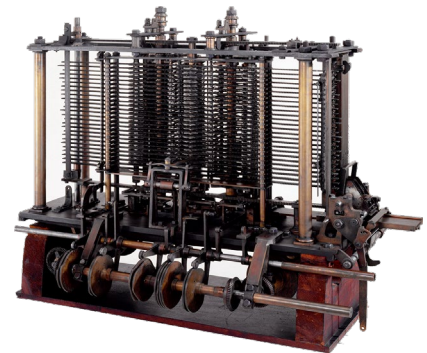
1. The Antikythera Mechanism

- The **Antikythera mechanism** is an ancient Greek device used to predict astronomical positions and eclipses. It is sometimes considered an early computing device because it performs calculations based on input (a date set by the user) and produces output (astronomical predictions).
- **Key takeaway:** It is similar to a computer because it processes input and produces output, but it is **purpose-built** for specific calculations. Unlike modern computers, it can't be reprogrammed to do different tasks.

2. Other Early Calculating Machines:



- **Pascaline** and **Stepped Reckoner** were early mechanical calculators that could add or subtract numbers automatically.



- **Babbage's Analytical Engine** was an early design for a general-purpose computer. Although it was never fully built, it was capable of being programmed to perform different tasks.

3. General-Purpose vs. Purpose-Built Devices:

- **Purpose-built devices** are designed to perform a specific function, such as a calculator or a clock.
 - A **general-purpose computer** can be programmed to perform many different tasks, making it much more versatile.
-

What Makes a Computer Different?

- **General-purpose computers** can execute a program, which allows them to transform into different types of machines. For example:
 - A computer running an astronomy program can work like the **Antikythera mechanism**.
 - A computer running a calculator program can perform similar tasks to the **Stepped Reckoner**.
 - A computer running a chess game can function like an electronic chess machine.
 - The key difference between general-purpose computers and other devices is their **programmability**. Once you install a different program, the computer can perform an entirely different task.
-

Understanding Programs and Software:

1. Programs vs. Data:

- **Programs** are sequences of instructions that tell the computer what to do.
- **Data** is information that is processed by programs. For example, a document, image, or video is data, while a program (like a word processor or image editor) lets you work with that data.

2. Apps:

- Programs that run on mobile devices are often called **apps**. While the word “app” is used for mobile programs, it’s essentially the same as any other program.

3. Operating Systems:

- An **operating system** (like Windows or macOS) is a special program that helps manage all other programs on a computer. It controls how programs run and how hardware (like the keyboard, screen, and storage) interacts with software.
-

Assessment and Recap:

1. What makes a computer different from a purpose-built device?

- A computer can execute a program, making it capable of performing many tasks. In contrast, purpose-built devices are designed for a single, fixed function.

2. What is the role of a program in a computer?

- A program provides the instructions that tell the computer what tasks to perform. Without a program, a computer cannot do anything.

3. Why are general-purpose computers so powerful?

- Because they can be reprogrammed to perform different tasks, they are incredibly versatile and can be used for anything from playing games to complex scientific calculations.

Lesson 2 - Under the Hood

Introduction

In this lesson, you will explore the hardware components that make up computing systems. Just as we use our senses and brain to perform tasks, computers rely on specific hardware to process information and execute programs. These components work together seamlessly to perform functions such as storing data, receiving input, and providing output.

Key Concepts

1. **Computing Systems:** A computing system is a combination of hardware and software that work together to execute programs and process data.
2. **Hardware:** The physical components of a computer system, including the processor, memory, storage devices, and input/output components.
3. **Architecture:** The overall design or structure of a computing system, which involves the arrangement and interaction of hardware components.

Learning Objectives Breakdown

1. Describe the Function of the Hardware Components in Computing Systems

Computing systems are made up of several essential hardware components, each with its own specific function:

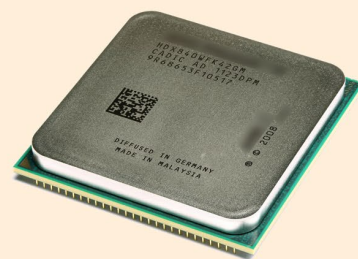
What does it do?

The **processor** is the component that **executes** program instructions.

An instruction may:

- Perform arithmetic or logic operations on data
- Perform input/output of data
- Control program flow

Terminology: The processor is commonly referred to as the **CPU** (central processing unit).



This is what the processor looks like in desktops and laptops.

Sometimes, the processor is integrated with other components, rather than being a separate component.

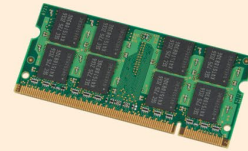
• **Processor (CPU):** Often referred to as the brain of the computer, the processor executes instructions and performs calculations. It controls the flow of data between various components of the system.

What does it do?

The main **memory** is the component that **stores** the programs and data **currently in use**.

Memory is **volatile**: its contents are lost when the power is off.

Terminology: The main memory is commonly referred to as **RAM** (random-access memory).



This is what the main memory looks like in desktops and laptops.

Sometimes, memory is integrated with other components, rather than being a separate component.

- **Memory:** Temporary storage used by the processor to store data and instructions that are actively being used or processed. There are two main types of memory:
 - **RAM (Random Access Memory):** Volatile memory, meaning it loses its data when power is turned off.
 - **Cache Memory:** A small, faster type of memory that stores frequently used data to speed up processing.

Question Name a few common storage devices.



Hard disk (HDD)



Solid-state drive (SSD)



SD card



USB stick



Optical disc

- **Storage:** Long-term data storage that retains information even when the computer is powered off. Examples include hard drives (HDD), solid-state drives (SSD), and optical drives.

Question Name a few common input devices.



Keyboard



Mouse



Camera



Microphone

Sensors

Question Name a few common output devices.



Screen



Speakers



Printer



Projector



LEDs



Motors

- **Input Devices:** Components that allow users to interact with the system by providing data. Examples include keyboards, mice, and microphones.
- **Output Devices:** Components that allow the system to communicate results back to the user. Examples include monitors, printers, and speakers.

2. Describe How Hardware Components Work Together to Execute Programs

- **Executing Programs:** When a user runs a program, the processor retrieves instructions from memory, decodes them, and executes them step by step. Data used in these instructions is stored temporarily in memory.
- **Communication Between Components:** The processor relies on input devices to receive data and output devices to display results. It uses memory to hold temporary data and storage to save data long-term.
- **Interaction Flow:**
 - **Input Devices** (e.g., keyboard) send data to the processor.
 - The **Processor (CPU)** processes the instructions and performs calculations.
 - **Memory (RAM)** stores data that needs to be accessed quickly while executing instructions.
 - The **Storage** stores data permanently.
 - **Output Devices** display or output the results of the executed program.

3. Recall That All Computing Systems Have a Similar Structure ('Architecture')

Despite differences in form (e.g., desktop computers, smartphones, and tablets), all computing systems share a similar structure, often referred to as **computer architecture**. This architecture involves the same core components (processor, memory, storage, input/output devices, communication) working together to execute programs.

Key Vocabulary

- **Computer:** A device that processes data and executes instructions.
 - **System:** A combination of components working together for a specific purpose.
 - **Device:** A hardware component used to perform tasks (e.g., keyboard, monitor).
 - **Program:** A set of instructions that a computer follows to perform a task.
 - **Instructions:** Specific commands that the processor executes.
 - **Data:** Information that is processed by a computer.
 - **Hardware:** The physical components of a computer system.
 - **Processor:** The component that executes instructions and controls the computer's operations.
 - **Memory:** Temporary storage that holds data and instructions while being processed.
 - **Storage:** Long-term storage for data and programs.
 - **Communication:** The exchange of data between devices, often over a network.
 - **Input and Output:** Devices that allow the computer to interact with users by receiving and displaying data.
 - **Architecture:** The design and structure of the components in a computing system.
-

Summary

In this lesson, we have learned about the fundamental hardware components that make up a computing system. The processor, memory, storage, input/output devices, and communication components all play vital roles in ensuring a computer can execute programs and process data efficiently. Understanding the architecture of computing systems helps us appreciate how these components work together in any device, whether it's a smartphone, laptop, or desktop computer.

Key Takeaways:

- All computing systems share a similar structure (architecture).
- The processor is the core component that controls and processes instructions.
- Memory provides temporary storage, while storage devices offer long-term data retention.
- Input and output devices allow interaction with the system, and communication components enable data exchange with other systems.

Review these components and their interactions to solidify your understanding of how computers work "under the hood."

Revision Guide: Lesson 3 - Orchestra Conductor



In some ways, the operating system is like the conductor of an orchestra.

Introduction

In this lesson, we'll take a deeper look at how the hardware components in a computer work together to execute programs, using real-life scenarios that you are familiar with. You'll also learn about the **Operating System**, which plays an essential role in managing all the hardware and software of a computing system, helping everything work smoothly.

Key Concepts

- **Hardware Components:** These include the processor, memory, storage, input/output devices, and communication components, which we learned about in the previous lesson.
- **Operating System (OS):** The software that helps control and manage all the hardware in a computing system. Think of it like the **orchestra conductor** that makes sure all the parts of the system work together smoothly.

Learning Objectives Breakdown

1. Analyse How the Hardware Components Work Together to Execute Programs

All the hardware components of a computing system work together to run programs and process data. Let's break down the role of each component in this process:

- **Processor (CPU):** The processor is like the "brain" of the computer. It fetches instructions from memory, processes them, and then executes them. It acts as the central control unit, directing operations.
- **Memory (RAM):** Memory temporarily stores data and instructions that are currently being used or processed by the CPU. This helps the processor access needed data quickly. Think of RAM as a workbench where the processor can organize and manipulate data on the fly.
- **Storage:** Storage devices (like hard drives or solid-state drives) hold data and programs even when the computer is turned off. It's like a filing cabinet for all your files and applications.

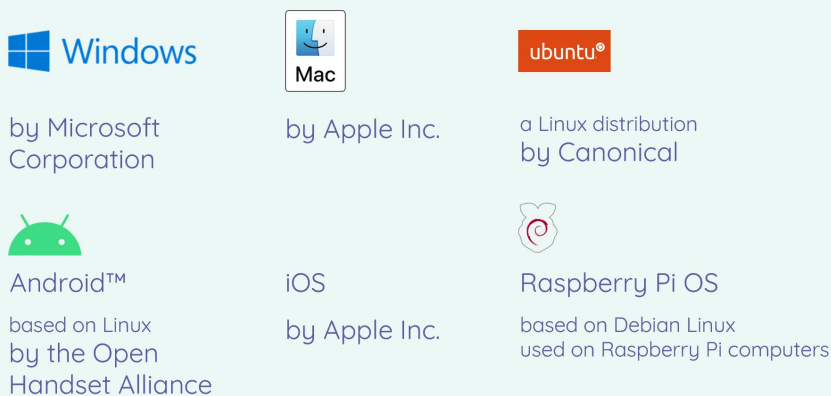
- **Input Devices:** These allow users to send data to the computer. Examples include the keyboard (for typing), mouse (for clicking), and microphone (for recording).
- **Output Devices:** These let the computer send data back to the user. Examples include monitors (for displaying information), printers (for printing documents), and speakers (for sound).
- **Communication Components:** These allow the computer to connect to other devices or networks. For example, Wi-Fi cards and Bluetooth adapters allow communication over the internet or between devices.

The Process:

1. **The CPU** starts by fetching instructions from **memory** (RAM).
2. The **processor** processes the instructions and may need to pull data from **storage** if it's not in memory.
3. If the program requires user input (e.g., typing or clicking), it will come through the **input devices**.
4. After processing, the results are sent to the **output devices** (e.g., showing text on a screen or playing a sound).
5. Communication components, such as a **network adapter**, might be used if the computer needs to connect to the internet or another device.

2. Define What an Operating System Is and Recall Its Role in Controlling Program Execution

The operating system: examples



An **Operating System (OS)** is the **software** that controls how all the hardware components of a computer work together. It acts as a **middleman** between the hardware and the software (programs you run).

- **What Does the OS Do?**
 - **Manage Hardware:** The OS makes sure the processor, memory, storage, input/output devices, and communication components all work smoothly together. It manages how data moves between these components.

- **Run Programs:** The OS manages program execution, making sure that each program gets the right amount of resources (such as memory or processing power) when it needs them.
- **Provide a User Interface:** The OS provides an interface (like Windows, macOS, or Android) that allows you to interact with the computer and control how the system operates.
- **Coordinate Input/Output:** The OS ensures that when you type on the keyboard or click the mouse, the computer correctly understands and responds to those actions.

Example: The Orchestra Conductor

Imagine a **computer system** as an **orchestra**. The **hardware components** (CPU, memory, storage, input/output devices) are the musicians, and the **Operating System** is the conductor who makes sure all the musicians play in time and follow the same rhythm. The **OS** ensures that every part of the system works together so that the program can run smoothly.

Summary

In this lesson, we learned how the different hardware components of a computing system work together to run programs and process data. The **Operating System** (OS) acts like a conductor of an orchestra, managing all the components and ensuring they work in harmony to execute programs successfully.

Key Takeaways:

- **The OS is essential** for managing hardware and software resources in a computer.
- **Hardware components** such as the processor, memory, storage, and input/output devices must all work together to execute programs.
- The **Operating System** helps these components work together efficiently, just like an orchestra conductor helps musicians play in time.

Make sure you understand the roles of both hardware components and the Operating System to see how they work together in a computing system!

Revision Guide: Lesson 4 - It's Only Logical

Introduction

In this lesson, we explore the connection between **logic** (the rules of reasoning) and **hardware** (the physical components that make up a computing system). Logical operations, like **NOT**, **AND**, and **OR**, are not just concepts used in programming but are fundamental to how computers process data at the hardware level.

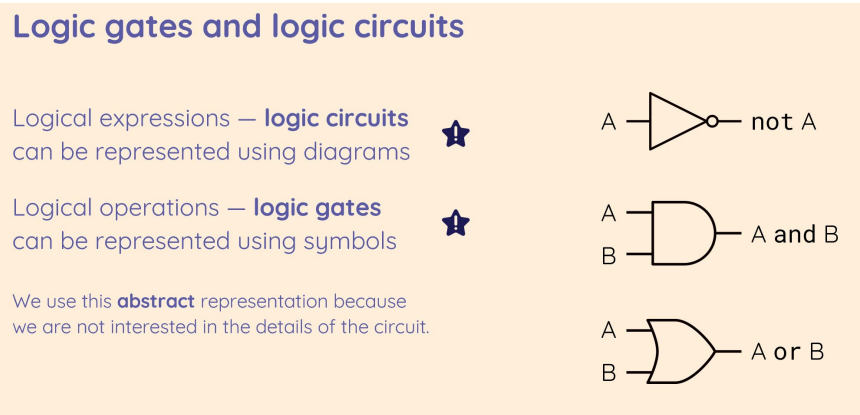
We'll learn how these logical operators work in **logic gates** and how logic circuits are built to perform tasks inside the computer, helping us understand how programs are executed by hardware.

Key Concepts

1. **Logical Operators:** These are the building blocks of logical expressions, used to make decisions in programming and hardware.
 - **NOT:** Reverses the value of a statement. If the value is **true**, it becomes **false**; if it's **false**, it becomes **true**.
 - **AND:** Both statements need to be **true** for the result to be **true**.
 - **OR:** If at least one of the statements is **true**, the result will be **true**.
 2. **Truth Values:** Logical expressions have two possible outcomes, often represented as **true** or **false**. These are also known as **truth values**.
 3. **Truth Tables:** A way of showing all possible outcomes of a logical expression based on different combinations of inputs (true or false).
 4. **Logic Gates:** These are physical devices that implement logical operators. Logic gates take one or more inputs and produce an output based on logical operations.
 5. **Logic Circuits:** A combination of logic gates that work together to process data and perform tasks. These circuits are the foundation of how computers perform calculations and execute programs.
-

Learning Objectives Breakdown

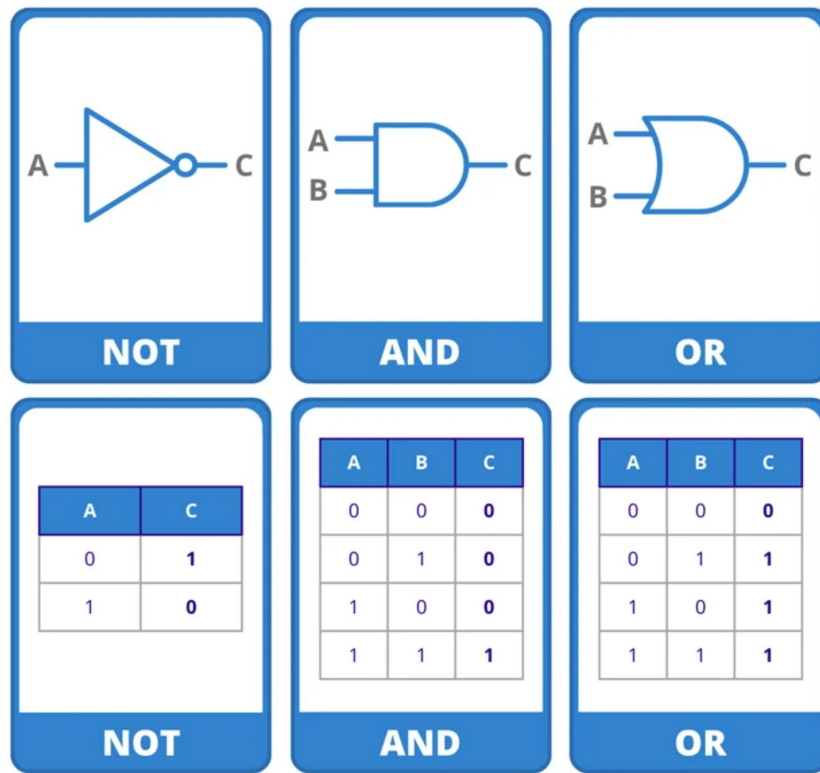
1. Describe the NOT, AND, and OR Logical Operators, and How They Are Used to Form Logical Expressions



Logical operators are the foundation of decision-making in both software and hardware. Here's how they work:

- **NOT:**
 - **Operation:** It changes the value of a statement. If the statement is **true**, the result becomes **false**, and vice versa.
 - **Example:**
 - NOT true = false
 - NOT false = true
- **AND:**
 - **Operation:** Both conditions need to be **true** for the result to be **true**.
 - **Example:**
 - true AND true = true
 - true AND false = false
 - false AND false = false
- **OR:**
 - **Operation:** If **at least one** condition is **true**, the result will be **true**.
 - **Example:**
 - true OR false = true
 - false OR false = false
 - true OR true = true

These operators are used to create **logical expressions** that help make decisions in programs and hardware systems.

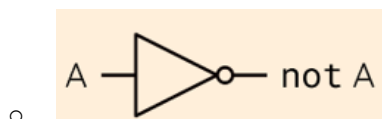


2. Use Logic Gates to Construct Logic Circuits, and Associate These with Logical Operators and Expressions

Logic Gates are physical devices inside the computer that perform logical operations. Each gate is connected to the hardware and responds to binary inputs (true or false, represented as 1 or 0) to produce an output.

- **NOT Gate:**

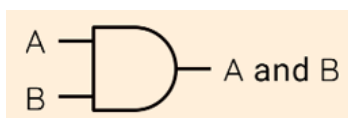
- **Operation:** Reverses the input value.



- **Example:** If the input is 1 (true), the output is 0 (false). If the input is 0 (false), the output is 1 (true).

- **AND Gate:**

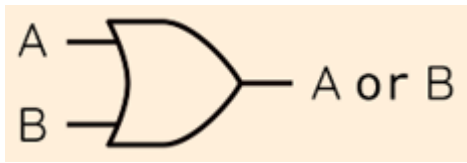
- **Operation:** Produces a 1 (true) only when both inputs are 1.



- **Example:**
 - $1 \text{ AND } 1 = 1$
 - $1 \text{ AND } 0 = 0$
 - $0 \text{ AND } 0 = 0$

- **OR Gate:**

- **Operation:** Produces a 1 (true) if at least one of the inputs is 1.



- **Example:**
 - $1 \text{ OR } 1 = 1$
 - $1 \text{ OR } 0 = 1$
 - $0 \text{ OR } 0 = 0$

By combining these gates, we can create **logic circuits** that perform more complex tasks. For example, combining **AND** and **OR** gates can help perform conditional checks or make decisions.

3. Recall that, Since Hardware is Built Out of Logic Circuits, Data and Instructions Alike Need to Be Represented Using Binary Digits

Since computers operate using binary (1s and 0s), data and instructions must also be represented in binary. This is because **logic gates** can only handle **binary values** (0 or 1).

- **Data Representation:** Everything in a computer (text, images, numbers) is converted into binary (1s and 0s) so that it can be processed by the logic circuits.
- **Instructions in Binary:** Programs are written in higher-level languages (like Python or Java), but these are converted into binary code so that the hardware can execute them.

Understanding how **binary** works is important because it helps explain how computers process and store data, and how logic circuits perform operations on that data.

Key Vocabulary

- **Logical Operators:** The basic building blocks of logic used to create logical expressions. Includes **NOT**, **AND**, and **OR**.
- **Logical Expressions:** A combination of logical operators used to evaluate conditions.
- **Truth Values:** The two possible outcomes of a logical expression: **true** (1) or **false** (0).
- **Truth Tables:** A table that shows all possible outcomes of a logical expression based on different inputs.
- **Logic Gates:** Physical devices that perform logical operations on binary inputs.
- **Logic Circuits:** Combinations of logic gates that perform more complex operations.
- **Binary Digits:** The representation of data using 1s and 0s. Also called **bits**.

Summary

In this lesson, we learned how **logic operators** like **NOT**, **AND**, and **OR** are used to create **logical expressions** that help computers make decisions. These operators are implemented through **logic gates**, which are the building blocks of **logic circuits** inside a computer. These circuits are used to perform all the tasks in a computer, from basic operations to running complex programs. Since computers work with binary (1s and 0s), both data and instructions are represented in binary form for processing.

Key Takeaways:

- **Logical operators** (NOT, AND, OR) form the basis of decision-making in computers.
- **Logic gates** perform simple operations, and **logic circuits** combine them to perform complex tasks.
- **Binary digits (1 and 0)** are used to represent all data and instructions in computers.

Revision Guide: Lesson 5 - Thinking Machines

Introduction

The idea of creating intelligent machines dates back to ancient myths, but it wasn't until the invention of the computer that the concept of artificial intelligence (AI) began to take shape. In 1950, Alan Turing posed the question, "Can machines think?" which sparked the field of AI. Over the past decade, advancements in **artificial intelligence** and **machine learning** (ML) have revolutionized many industries, from healthcare to entertainment. This lesson will explore what AI and ML are, how they work, and their real-world applications.

Key Concepts

1. **Artificial Intelligence (AI):** AI refers to machines or software that can perform tasks that would normally require human intelligence. These tasks might include problem-solving, decision-making, language understanding, and pattern recognition.
 2. **Machine Learning (ML):** ML is a type of AI where machines learn from data rather than being explicitly programmed. This allows systems to improve over time as they are exposed to more data.
 3. **Training a Model:** Machine learning involves training models with data so they can recognize patterns and make predictions. Training involves collecting data, testing the model, and refining it based on performance.
 4. **Ethical Considerations:** As AI and ML become more integrated into everyday life, questions about their fairness, transparency, and impact on society arise. Ethical considerations involve ensuring that AI systems are used responsibly and do not cause harm.
-

Learning Objectives Breakdown

1. Provide Broad Definitions of 'Artificial Intelligence' and 'Machine Learning'

- **Artificial Intelligence (AI):**
 - AI refers to **machines** that are designed to perform tasks typically requiring human intelligence. For example, AI can play games like chess, recognize faces in photos, or help autonomous cars drive themselves.
 - **Examples of AI:**
 - **Voice assistants** (like Siri or Alexa)
 - **Autonomous vehicles** (self-driving cars)
 - **Recommendation systems** (like Netflix or YouTube suggesting videos)
 - **Medical diagnosis** (AI helping doctors identify diseases in scans)
 - **Machine Learning (ML):**
 - ML is a **subfield of AI** that focuses on **training machines** to learn from data. Instead of writing detailed rules for every task, a machine can learn from patterns in data to make predictions or decisions.
 - **Examples of ML:**
 - **Email filtering** (spam filters)
 - **Facial recognition** (unlocking your phone with your face)
 - **Speech recognition** (transcribing spoken words into text)
 - **Predicting trends** (like stock prices or weather forecasts)
-

2. Identify Examples of Artificial Intelligence and Machine Learning in the Real World

- **Artificial Intelligence Examples:**
 - **Smartphones:** AI powers features like **voice recognition** and **face recognition** on phones.
 - **E-commerce Websites:** AI helps online stores **recommend products** based on your browsing history (e.g., Amazon's product recommendations).
 - **Healthcare:** AI is used to analyze **medical images** and assist doctors in diagnosing conditions.
- **Machine Learning Examples:**
 - **Social Media:** Platforms like Facebook and Instagram use **ML** to suggest posts, ads, and people you may know based on your activity.
 - **Autonomous Vehicles:** Self-driving cars use **ML algorithms** to learn from sensors and cameras to navigate roads safely.

- **Video Games:** AI-controlled characters (non-player characters, NPCs) in games learn from players' actions and adapt to provide more realistic challenges.
-

3. Describe the Steps Involved in Training Machines to Perform Tasks (Gathering Data, Training, Testing)

Training a machine to perform a task is a process that involves several key steps:

- **Gathering Data:**

- The first step is to collect **data** that the machine will use to learn. For example, if we want to train a machine to recognize cats in photos, we need a large set of images labeled as “cat” and “not cat.”
- The more **diverse** and **accurate** the data, the better the model will perform.

- **Training:**

- During the training phase, the machine learns from the data by **identifying patterns**. The machine adjusts its internal settings (called parameters) to improve its ability to make predictions.
- For example, an AI model might learn what a cat looks like by studying thousands of cat images and recognizing common features like ears, whiskers, and fur.

- **Testing:**

- Once the machine has been trained, it is tested on new data it has never seen before. This helps evaluate how well the model can make predictions on **unseen data**.
 - If the model performs well, it can be deployed. If not, the data and training process might need to be improved.
-

4. Describe How Machine Learning Differs from Traditional Programming

- **Traditional Programming:**

- In traditional programming, a programmer writes **explicit instructions** for the computer to follow. For example, if you want a program to sort a list of numbers, you would write a specific algorithm to compare each number and sort them.
- This is a **rule-based** approach, where the programmer defines how the program works.

- **Machine Learning:**

- In contrast, machine learning allows the machine to **learn** how to perform tasks from data rather than relying on explicit instructions. The computer is trained on data and can adapt its behavior over time.
- For example, instead of programming a computer to identify cats in photos, you provide labeled images, and the machine learns what features make up a cat.

5. Associate the Use of Artificial Intelligence with Moral Dilemmas

As AI systems become more powerful and widespread, they raise important **ethical questions**:

- **Bias in AI:** AI models can unintentionally **reflect biases** in the data they are trained on. For example, an AI trained on biased hiring data may unfairly favor certain groups over others.
- **Privacy:** AI systems, like facial recognition or surveillance cameras, can raise concerns about **privacy** and how data is collected, stored, and used.
- **Decision-Making:** As AI makes more decisions, questions arise about **who is responsible** when an AI system makes a mistake, such as in self-driving cars causing accidents.
- **Job Displacement:** AI and automation may lead to **job losses** in certain industries, causing concern about its impact on the workforce.

Key Vocabulary

- **Artificial Intelligence (AI):** Machines designed to perform tasks that require human-like intelligence.
- **Machine Learning (ML):** A type of AI where machines learn from data rather than following fixed instructions.
- **Training:** The process of teaching a machine to recognize patterns in data.
- **Testing:** Evaluating a machine's ability to make predictions with new, unseen data.
- **Programming:** Writing explicit instructions for a computer to follow.
- **Ethical Considerations:** The moral issues surrounding the use of AI, such as bias, privacy, and accountability.

Summary

In this lesson, we explored the fascinating fields of **Artificial Intelligence** and **Machine Learning**. We learned that AI allows machines to perform tasks requiring human intelligence, and that ML teaches machines to learn from data instead of following predefined rules. As these technologies develop, they offer exciting possibilities but also present important ethical challenges. Understanding AI and ML is essential for navigating the future of technology responsibly.

Key Takeaways:

- **AI** allows machines to think and make decisions, while **ML** enables them to learn from data.
- **Training a machine** involves gathering data, teaching it, and testing it to see how well it performs.

- **Machine learning** differs from traditional programming by learning from data rather than following explicit instructions.
- **Ethical dilemmas** around AI include bias, privacy, and accountability.

With AI and ML becoming part of our everyday lives, it's important to understand how they work and their impact on society!