



SELÇUK ÜNİVERSİTESİ
HUĞLU MESLEK YÜKSEKOKULU

BİLGİSAYAR TEKNOLOJİLERİ BÖLÜMÜ
MESLEKİ YABANCI DİL DERS NOTLARI

İçindekiler

1.COMPUTER HARDWARE.....	4
1.1.Computers and Their Uses	4
1.2.About Comuputer Hardware	4
2.OPERATING SYSTEMS	6
2.1.CPU Scheduling	6
2.2.Scheduling Algorithms.....	6
2.2.1. First-Come, First-Served Scheduling.....	7
2.2.2. Shortest-Job-First Scheduling	7
2.2.3.Priority Scheduling.....	7
2.2.4.Round-Robin Scheduling	8
3.EMBEDDED SYSTEM	8
3.1. An Overview of Embedded Systems.....	8
3.2. Differences Between an Embedded Controller and a Personal Computer	9
4.CYBER SECURITY	9
4.1. Introduction to Malware Analysis	9
4.2. What Is Malware?	10
4.3. What Is Malware Analysis?.....	11
4.4. Types Of Malware Analysis.....	11
5.DATABASE SYSTEMS.....	12
5.1. Introduction to Database	12
5.2.Database Engine.....	12
6.WEB DESIGN.....	13
6.1.Web Design Framework	13
6.2. Information Architecture.....	14
6.3. Web Navigation.....	14
6.4. Search Bar	14
6.5. Content Strategy	15
6.6. Visual Elements	15
7.COMPUTER NETWORKS	16
7.1.Types of Networks.....	17
7.2. Basic Components of a Network	18
8.PROGRAMMING	19
8.1.Creation of C#.....	19
8.1.1.What is the .NET Framework?.....	19
8.2.PYTHON Programming Language	20
8.3.Object Oriented Programing	21

9.ARTIFICIAL INTELLIGENCE.....	22
9.1. Understanding Artificial Intelligence	22
9.2. Machine Learning: The Essence of AI	23
9.3. Deep learning	24
9.3.1. Definition of Deep learning	24
9.3.2. History of Deep learning.....	25
10.MICROSOFT WINDOWS.....	26
10.1. Power Shell.....	26
10.2. What Can You Use PowerShell For?	26
10.3. Windows Commands.....	26
11.CLOUD COMPUTING.....	27
11.1. Types of Cloud Computing Services.....	28
12. Augmented Reality and Virtual Reality.....	28
12.1. Understanding Augmented Reality	28
12.2. Exploring Virtual Reality	29
12.3. Differentiating between AR and VR technologies:	30

1.COMPUTER HARDWARE

1.1.Computers and Their Uses

Computers are everywhere! From office desks to kitchen tables, from library tables to people's laps in coffee shops, the screen and keyboard combination is a familiar sight. But if you look more carefully, you'll discover that computers exist in even more places than you first realize. Cash registers in stores have computers inside that calculate prices and help manage inventory. Most cars produced today have diagnostic computers to help find problems and improve performance. Cell phones contain computers, and wristwatches often rely on simple computers for stopwatch and calendar functions. Since the 1970s, computers have rapidly reshaped personal and business life as we know it. Factories and industrial companies often use computer-controlled machinery. Professional occupations such as law, engineering, medicine, and finance use computers almost universally. Many workers who once had little use for technology now interact with computers almost every minute of the workday. At home, people use computers to read the news, play games, and keep in touch with family and friends. News from around the world is instantly available. More and more people have international friends and colleagues, and they can use computers to communicate.

1.2.About Computer Hardware

In today's digital era, computers have become an integral part of our lives. They are used for a wide range of tasks such as communication, entertainment, data processing, and business operations. To better understand how computers work and enable ourselves to troubleshoot issues when they arise, it is essential to be familiar with the basic hardware components that make up a computer system. In this article, we will discuss the various essential computer hardware components and their functions.

Central Processing Unit (CPU):

The CPU serves as the brain of the computer. It is responsible for executing instructions and performing calculations. The CPU comprises two main parts: the control unit and the arithmetic logic unit (ALU). The control unit manages the flow of data and instructions between computer components, while the ALU performs arithmetic and logical operations on data.

Random Access Memory (RAM):

RAM is a type of temporary memory storage that allows computers to store and quickly access data needed by the CPU to perform tasks. Since RAM is volatile memory, it retains its data only when it has power; once the power is turned off, all data stored in RAM is lost.

Motherboard:

The motherboard serves as a central hub that connects all computer hardware components. It provides essential pathways for data communication between parts such as the CPU, RAM, storage devices, and peripherals like keyboard, mouse or monitor.

Storage Devices:

Computer storage devices include hard disk drives (HDDs), solid-state drives (SSDs), and optical drives such as CD/DVD/Blu-ray drives. Storage devices store user files, programs, and operating systems that are needed for a computer to function properly.

Graphics Processing Unit (GPU):

The GPU or graphics card is responsible for rendering images, videos, and animations that appear on screen. The GPU processes this visual data faster than a CPU would do allowing for smooth visual experience in gaming, video editing, and other graphics-intensive tasks.

Power Supply Unit (PSU):

The PSU provides electrical power for all computer components and converts alternating current (AC) from the power outlet to direct current (DC) required by the computer circuitry.

Cooling System:

The cooling system dissipates heat produced by the computer's components during operation, ensuring stable performance and preventing overheating. This system can include air cooling with fans, liquid cooling with water or other fluids circulated through specialized tubes, or even passive cooling with heat sinks.

Peripherals:

These include input and output devices like a keyboard, mouse, monitor, printer, and speakers which facilitate easier interaction between users and the computer.

Understanding the basic computer hardware components is essential for troubleshooting and enhancing system performance. Each hardware component plays a crucial role in ensuring that your computer runs efficiently. With this knowledge in hand, you'll be better prepared to solve issues or choose the perfect components when upgrading or building a new computer system.

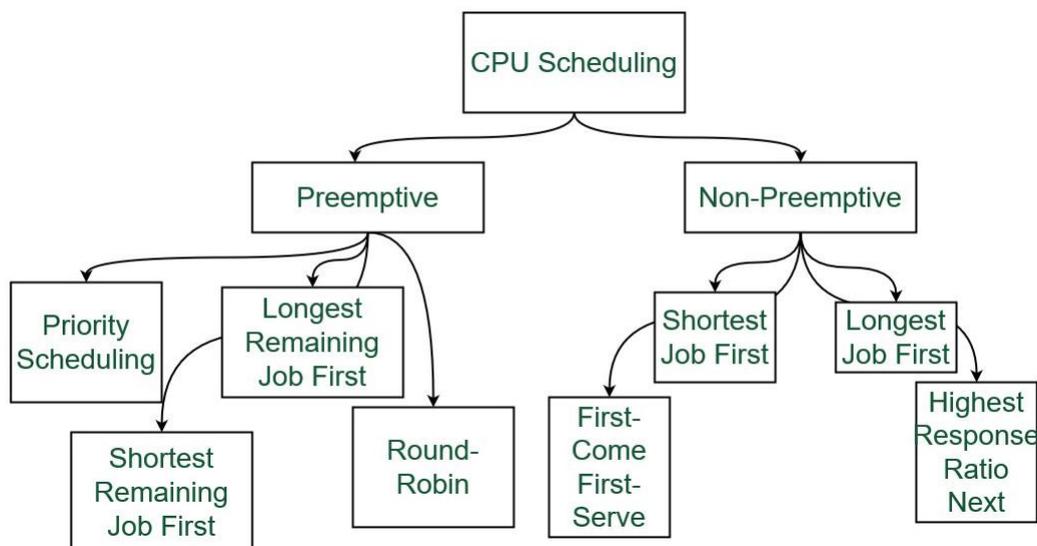
2. OPERATING SYSTEMS

2.1. CPU Scheduling

CPU scheduling is the basis of multiprogrammed operating systems. By switching the CPU among processes, the operating system can make the computer more productive. In this chapter, we introduce basic CPU-scheduling concepts and present several CPU-scheduling algorithms. We also consider the problem of selecting an algorithm for a particular system. In a single-processor system, only one process can run at a time. Others must wait until the CPU is free and can be rescheduled. The objective of multiprogramming is to have some process running at all times, to maximize CPU utilization. The idea is relatively simple. A process is executed until it must wait, typically for the completion of some I/O request. In a simple computer system, the CPU then just sits idle. All this waiting time is wasted; no useful work is accomplished. With multiprogramming, we try to use this time productively. Several processes are kept in memory at one time. When one process has to wait, the operating system takes the CPU away from that process and gives the CPU to another process. This pattern continues. Every time one process has to wait, another process can take over use of the CPU. Scheduling of this kind is a fundamental operating-system function. Almost all computer resources are scheduled before use. The CPU is, of course, one of the primary computer resources. Thus, its scheduling is central to operating-system design.

2.2. Scheduling Algorithms

CPU scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU. There are many different CPU-scheduling algorithms. In this section, we describe several of them.



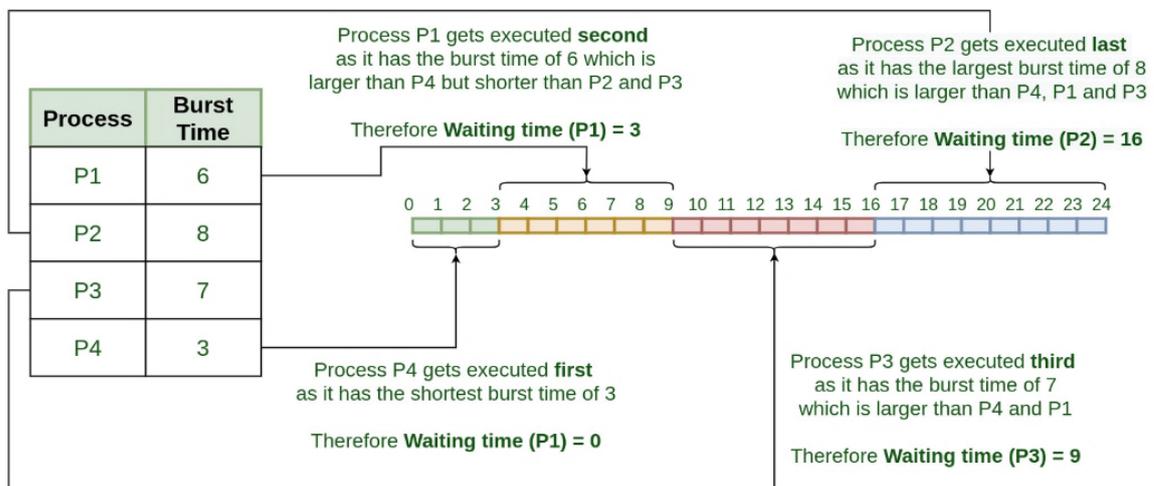
2.2.1. First-Come, First-Served Scheduling

By far the simplest CPU-scheduling algorithm is the first-come, first-served (FCFS) scheduling algorithm. With this scheme, the process that requests the CPU first is allocated the CPU first. The implementation of the CFS policy is easily managed with a FIFO queue. When a process enters the ready queue, its PCB is linked onto the tail of the queue. When the CPU is free, it is allocated to the process at the head of the queue. The running process is then removed from the queue. The code for FCFS scheduling is simple to write and understand.

2.2.2. Shortest-Job-First Scheduling

A different approach to CPU scheduling is the shortest-job-first (SJF) scheduling algorithm. This algorithm associates with each process the length of the process's next CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If the next CPU bursts of two processes are the same, FCFS scheduling is used to break the tie. Note that a more appropriate term for this scheduling method would be the shortest-next-CPU-burst algorithm, because scheduling depends on the length of the next CPU burst of a process, rather than its total length. We use the term SJF because most people and textbooks use this term to refer to this type of scheduling.

Shortest Job First (SJF) Scheduling Algorithm



2.2.3. Priority Scheduling

The SJF algorithm is a special case of the general priority-scheduling algorithm. A priority is associated with each process, and the CPU is allocated to the process with the highest priority. Equal-priority processes are scheduled in FCFS order. An SJF algorithm is simply a priority algorithm where the priority (p) is the inverse of the (predicted) next CPU burst. The larger the CPU burst, the lower the priority, and vice versa.

2.2.4.Round-Robin Scheduling

The round-robin (RR) scheduling algorithm is designed especially for time-sharing systems. It is similar to FCFS scheduling, but preemption is added to enable the system to switch between processes. A small unit of time, called a time quantum or time slice, is defined. A time quantum is generally from 10 to 100 milliseconds in length. The ready queue is treated as a circular queue.

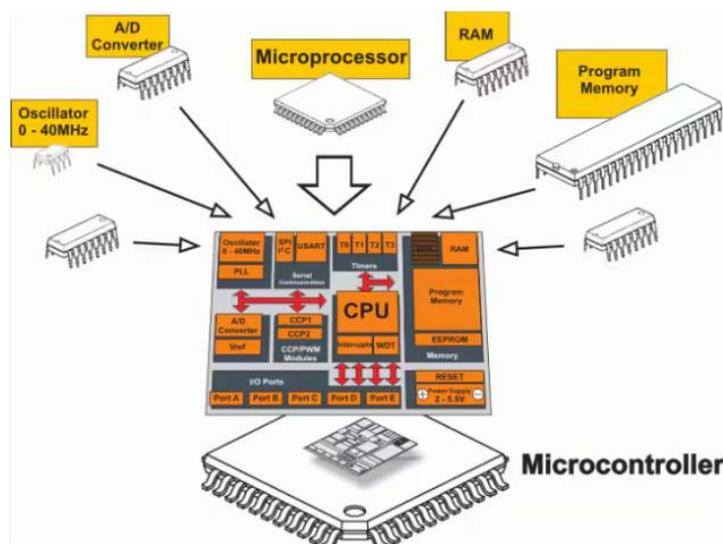
The CPU scheduler goes around the ready queue, allocating the CPU to each process for a time interval of up to 1 time quantum. To implement RR scheduling, we again treat the ready queue as a FIFO queue of processes. New processes are added to the tail of the ready queue. The CPU scheduler picks the first process from the ready queue, sets a timer to interrupt after 1 time quantum, and dispatches the process.

One of two things will then happen. The process may have a CPU burst of less than 1 time quantum. In this case, the process itself will release the CPU voluntarily. The scheduler will then proceed to the next process in the ready queue. If the CPU burst of the currently running process is longer than 1 time quantum, the timer will go off and will cause an interrupt to the operating system. A context switch will be executed, and the process will be put at the tail of the ready queue. The CPU scheduler will then select the next process in the ready queue.

3.EMBEDDED SYSTEM

3.1. An Overview of Embedded Systems

An embedded system is typically a design that uses the power of a small microcontroller, like the Microchip AVR microcontroller (MCU). These microcontrollers combine a microprocessor unit (like the CPU in a personal computer) with some additional circuits called peripherals, plus some additional circuits, on the same chip to make a small control module requiring few other external devices. This single device can then be embedded into other electronic and mechanical devices for low-cost digital control.



3.2. Differences Between an Embedded Controller and a Personal Computer

The main difference between an embedded controller and a personal computer is that the embedded controller is dedicated to one specific task or set of tasks. A personal computer is designed to run many different types of programs and to connect to many different external devices. An embedded controller has a single program and, as a result, can be made cheaply to include just enough computing power and hardware to perform that dedicated task.

A personal computer has a relatively expensive generalized central processing unit (CPU) at its heart with many other external devices (memory, disk drives, video controllers, network interface circuits, etc.). An embedded system has a low-cost MCU for its intelligence, has many peripheral circuits on the same chip and has relatively few external devices.

Often an embedded system is an invisible part, or sub-module of another product, such as a cordless drill, refrigerator or garage door opener. The controller in these products does a tiny portion of the function of the whole device. The controller adds low-cost intelligence to some of the critical sub-systems in these devices.

An example of an embedded system is a smoke detector. Its function is to evaluate signals from a sensor and sound an alarm if the signals indicate the presence of smoke. A small program in the smoke detector either runs in an infinite loop, sampling the signal from the smoke sensor, or lies dormant in a low-power “Sleep” mode, being awakened by a signal from the sensor. The program then sounds the alarm. The program would possibly have a few other functions, such as a user test function and a low battery alert.

While a personal computer with a sensor and audio output could be programmed to do the same function, it would not be a cost-effective solution (nor would it run on a nine-volt battery, unattended for years). Embedded designs use inexpensive microcontrollers to put intelligence into the everyday things in our environment, such as smoke detectors, cameras, cell phones, appliances, automobiles, smart cards and security systems.

4. CYBER SECURITY

4.1. Introduction to Malware Analysis

The number of cyber attacks is undoubtedly on the rise, targeting government, military, public and private sectors. These cyber attacks focus on targeting individuals or organizations with an effort to extract valuable information. Sometimes, these cyber attacks are allegedly linked to cybercrime or state-sponsored groups, but may also be carried out by individual groups to achieve their goals. Most of these cyber attacks use malicious software (also called malware) to infect their targets. Knowledge, skills, and tools required to analyze malicious software are essential to detect, investigate and defend against such attacks.

4.2. What Is Malware?

Malware is a code that performs malicious actions; it can take the form of an executable, script, code, or any other software. Attackers use malware to steal sensitive information, spy on the infected system, or take control of the system. It typically gets into your system without your consent and can be delivered via various communication channels such as email, web, or USB drives.

The following are some of the malicious actions performed by malware:

- Disrupting computer operations
- Stealing sensitive information, including personal, business, and financial data
- Unauthorized access to the victim's system
- Spying on the victims
- Sending spam emails
- Engaging in distributed-denial-of-service attacks (DDOS)
- Locking up the files on the computer and holding them for ransom

Malware is a broad term that refers to different types of malicious programs such as trojans, viruses, worms, and rootkits. While performing malware analysis, you will often come across various types of malicious programs; some of these malicious programs are categorized based on their functionality and attack vectors as mentioned here:

Virus or Worm: Malware that is capable of copying itself and spreading to other computers. A virus needs user intervention, whereas a worm can spread without user intervention.

Trojan: Malware that disguises itself as a regular program to trick users to install it on their systems. Once installed, it can perform malicious actions such as stealing sensitive data, uploading files to the attacker's server, or monitoring webcams.

Backdoor / Remote Access Trojan (RAT): This is a type of Trojan that enables the attacker to gain access to and execute commands on the compromised system. Adware: Malware that presents unwanted advertisements (ads) to the user. They usually get delivered via free downloads and can forcibly install software on your system.

Botnet: This is a group of computers infected with the same malware (called bots), waiting to receive instructions from the command-and-control server controlled by the attacker. The attacker can then issue a command to these bots, which can perform malicious activities such as DDOS attacks or sending spam emails.

Information stealer: Malware designed to steal sensitive data such as banking credentials or typed keystrokes from the infected system. Some examples of these malicious programs include key loggers, spyware, sniffers, and form grabbers.

Ransomware: Malware that holds the system for ransom by locking users out of their computer or by encrypting their files.

Rootkit: Malware that provides the attacker with privileged access to the infected system and conceals its presence or the presence of other software.

Downloader or dropper: Malware designed to download or install additional malware components.

Classifying malware based on their functionalities may not always be possible because a single malware can contain multiple functionalities, which may fall into a variety of categories mentioned previously. For example, malware can include a worm component that scans the network looking for vulnerable systems and can drop another malware component such as a backdoor or a ransomware upon successful exploitation.

4.3. What Is Malware Analysis?

Malware analysis is the study of malware's behavior. The objective of malware analysis is to understand the working of malware and how to detect and eliminate it. It involves analyzing the suspect binary in a safe environment to identify its characteristics and functionalities so that better defenses can be built to protect an organization's network.

4.4. Types Of Malware Analysis

To understand the working and the characteristics of malware and to assess its impact on the system, you will often use different analysis techniques. The following is the classification of these analysis techniques:

Static analysis: This is the process of analyzing a binary without executing it. It is easiest to perform and allows you to extract the metadata associated with the suspect binary. Static analysis might not reveal all the required information, but it can sometimes provide interesting information that helps in determining where to focus your subsequent analysis efforts.

Dynamic analysis (Behavioral Analysis): This is the process of executing the suspect binary in an isolated environment and monitoring its behavior. This analysis technique is easy to perform and gives valuable insights into the activity of the binary during its execution. This analysis technique is useful but does not reveal all the functionalities of the hostile program.

Code analysis: It is an advanced technique that focuses on analyzing the code to understand the inner workings of the binary. This technique reveals information that is not possible to determine just from static and dynamic analysis. Code analysis is further divided into Static code analysis and Dynamic code analysis. Static code analysis involves disassembling the suspect binary and looking at the code to understand the program's behavior, whereas Dynamic code analysis involves debugging the suspect binary in a controlled manner to understand its functionality. Code analysis requires an understanding of the programming language and operating system concepts.

Memory analysis (Memory forensics): This is the technique of analyzing the computer's RAM for forensic artifacts. It is typically a forensic technique, but integrating it into your malware analysis will assist in gaining an understanding of the malware's behavior after infection. Memory analysis is especially useful to determine the stealth and evasive capabilities of the malware.

5.DATABASE SYSTEMS

5.1. Introduction to Database

A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient. Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results. Because information is so important in most organizations, computer scientists have developed a large body of concepts and techniques for managing data. These concepts and techniques form the focus of this book. This chapter briefly introduces the principles of database systems.

5.2.Database Engine

A database system is partitioned into modules that deal with each of the responsibilities of the overall system. The functional components of a database system can be broadly divided into the storage manager, the query processor components, and the transaction management component.

The storage manager is important because databases typically require a large amount of storage space. Corporate databases commonly range in size from hundreds of gigabytes to terabytes of data. A gigabyte is approximately 1 billion bytes, or 1000 megabytes (more precisely, 1024 megabytes), while a terabyte is approximately 1 trillion bytes or 1 million megabytes (more precisely, 1024 gigabytes). The largest enterprises have databases that reach into the multi-petabyte range (a petabyte is 1024 terabytes). Since the main memory of computers cannot store this much information, and since the contents of main memory are lost in a system crash, the information is stored on disks. Data are moved between disk storage and main memory as needed. Since the movement of data to and from disk is slow relative to the speed of the central processing unit, it is imperative that the database system structure the data so as to minimize the need to move data between disk and main memory. Increasingly, solid-state disks (SSDs) are being used for database storage. SSDs are faster than traditional disks but also more costly.

The query processor is important because it helps the database system to simplify and facilitate access to data. The query processor allows database users to obtain good performance while being able to work at the view level and not be burdened with understanding the physical-level details of the implementation of the

system. It is the job of the database system to translate updates and queries written in a nonprocedural language, at the logical level, into an efficient sequence of operations at the physical level.

6.WEB DESIGN

Getting started in web design? You have landed at the right address. This is a complete guide to Web Design which will help you learn everything needed to create a website so that you design your web presence to add value to your specific business and stand out from millions of other templates.

Web Design, in the simplest term, is the process of planning, conceptualizing, and arranging content intended to mark your business or individual presence on the Internet. In today’s world, we design has moved beyond just the websites to web apps, mobile apps, and user interface design. The latest web design focuses on making your website look appealing along with offering simplified user experience.

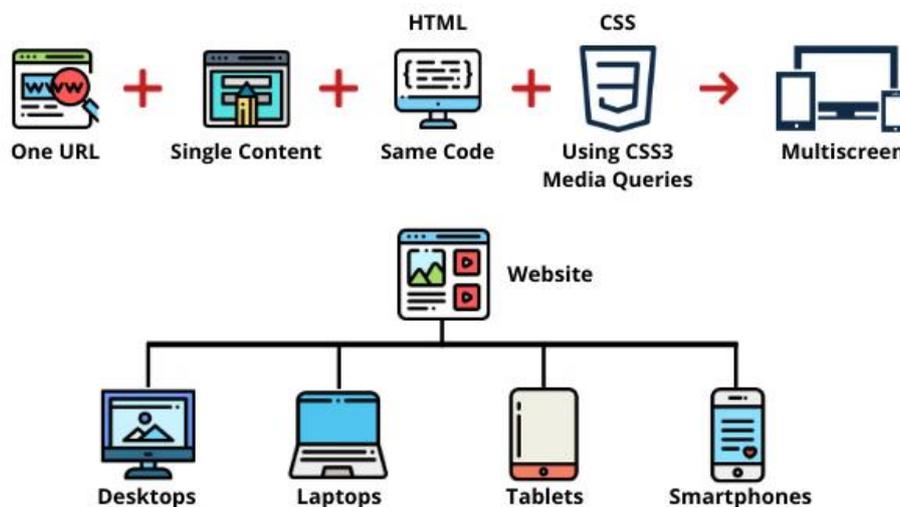
Web design is a compilation of several skill sets, including creating engaging content, designing visually appealing images and color schemes, optimization for search engines, optimization for conversions, branding and so much more.

“There are three responses to a piece of design — yes, no, and WOW! Wow is the one to aim for.”- Milton Glaser

6.1.Web Design Framework

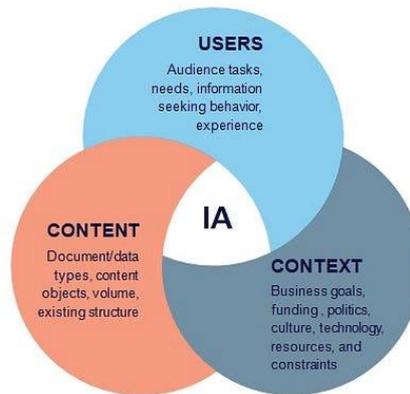
It is the UI design framework or wireframe of the website that speeds up the development of a web design process. Developers use frameworks to create beautiful and responsive designs. A framework, ideally, should:

- Be an open-source
- Be customizable as per the project need
- Work on all browsers
- Enhance all HTML elements



6.2. Information Architecture

Organizing the information that you intend to put on your website in an organized manner is called Information Architecture. The need for a full-proof IA exists to help users to navigate a complex set of information. While developing IA, one has to keep a tab on everything, right from creating the user journey to future-proofing the process.



To put it simply, IA is the “art and science of organizing and labeling data including websites, intranets, online communities, software, books and other mediums of information, to support usability”.

6.3. Web Navigation

A website’s conversions, sales, and bounce rates heavily rely upon the navigation and its ease. Creating, clear, hierarchical website navigation that leads your visitors to find what they want instantly increases your web visit rates, thereby improving your SEO rankings too.

It’s imperative to link all the menus and buttons on a website with the navigational elements of your website. Internal and external links should have clear differences. Ensure that all internal links open in the same tab while external links can lead the user to a new window or tab.

While designing navigation, a developer needs to focus upon:

1. Choosing a pattern based on the user’s needs
2. Assigning priority levels to the navigation options
3. Giving visibility to the navigation

Always double-check the functionality of the links added to your website. 404 errors or broken pages are a huge disappointment for the users.

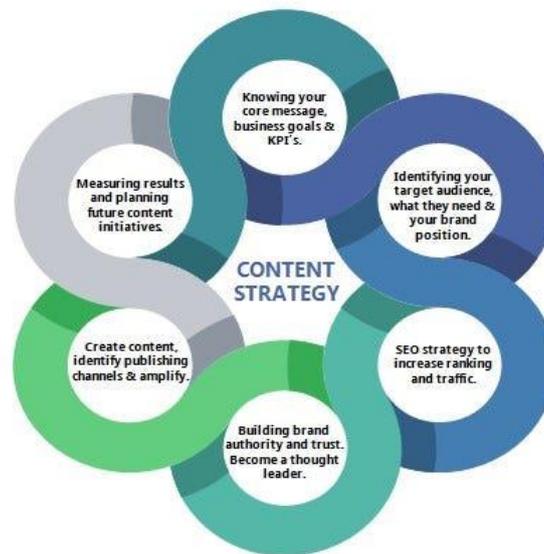
6.4. Search Bar

With users getting picky with time, the value of the search bar has become unparalleled. Some users to your website might visit it looking for a particular piece of info, product, or images. They just type the query in the search bar and voila they get what they are looking for. While designing a search box, one must remember to:

- Place the search bar at a visible spot
- Check the size of the input box: shouldn't be too big or too small
- Add a search box on every page



6.5. Content Strategy



Defining a content strategy aligned with your web presence goals is perhaps the most important part of web designing. We've all heard "content is king" over and over again. Content is a mix of visuals and text matter that the user sees on a website. Sharing common grounds with user experience (UX), interface design, web development, SEO, content marketing, public relations, and traditional "offline" marketing, content can make or break your objectives.

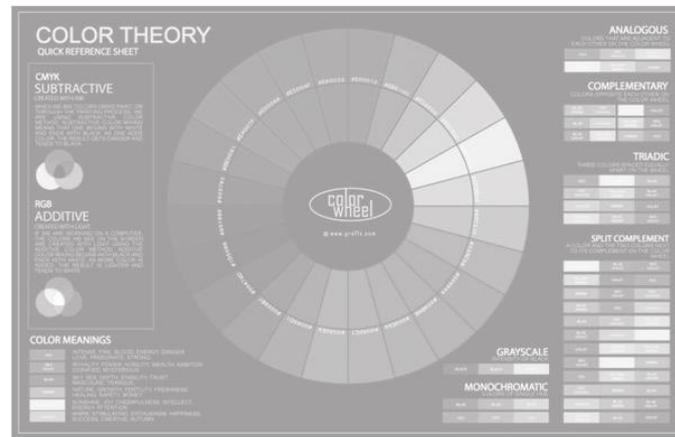
An effective content strategy needs to be mission-specific, relevant, comprehensive, and adaptable. Videos, blog posts, social media messages, podcasts, press releases, white papers, etc. are a part of a content strategy and must sync with your web goals.

6.6. Visual Elements

The most crucial component of web design is the visual appeal it brings to the forefront. Just like the real world, first impressions are critical even in the virtual world, so you would prefer to woo your audience the moment your page loads. This simply means that your design should be simple, familiar, intuitive, clean, and accessible. Using ample whitespace (or padding and margins) to give the elements of your site room to breathe, and grid-based designs to bring an organized, structured look to your website can. Of course, strong

photography and graphics provide comprehensive information to your text but ensure that the images complement each other and go with your business branding.

Choosing a centralized color scheme for your website is also an integral part of web design. Most web-design beginners start with creating brand-specific color palettes as it's this element that forms opinions for your website in your visitor's brains.



Most people don't know this and often choose to even notice the power of typography and fonts. When designing a website, it's imperative to choose easy-to-read font pairings that complement the design.



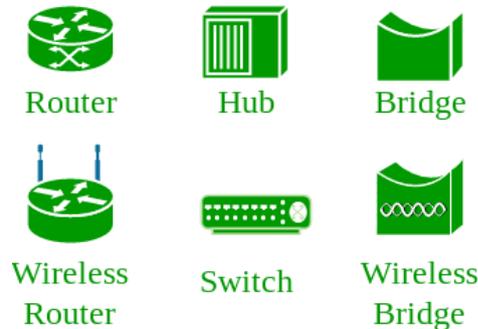
The world has come leaps and bounds in the field typeface options. Typographs are now crafted for specific screens and designed to enhance legibility. Just as printed fonts trick the eye to increase legibility, like ink traps, good web fonts will do the same. Designers can now take a break from Times New Roman and get access to typefaces that are web-safe and accessible. Google Fonts, Font Library, are a few that offer easy integration and a wide range of choices.

7.COMPUTER NETWORKS

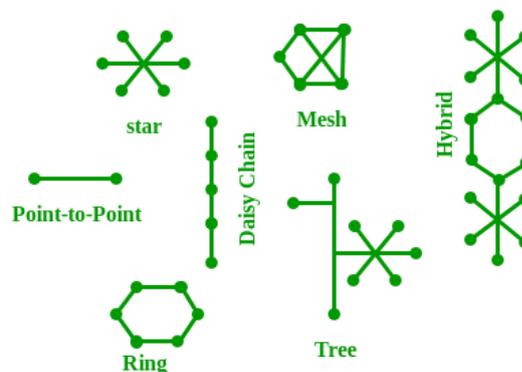
The integral role of computer networking in the modern world can't be overstated. As the vital structure underpinning the Internet and our global communication infrastructure, it enables rapid information exchange and resource sharing, irrespective of geographical distance. This article delves into the core concepts of networking, shedding light on its diverse types, the fundamental components, network protocols, and other integral aspects that contribute to its functionality.

It is the interconnection of multiple devices, generally termed as Hosts connected using multiple paths for the purpose of sending/receiving data or media.

There are also multiple devices or mediums which help in the communication between two different devices which are known as Network devices. Ex: Router, Switch, Hub, Bridge.



The layout pattern using which devices are interconnected is called as network topology. Such as Bus, Star, Mesh, Ring, Daisy chain.



7.1.Types of Networks

Local Area Networks (LAN): A LAN connects computers within a limited geographical area such as a home, office, or a group of adjacent buildings. For instance, when you connect your laptop, smartphone, and smart TV to the same Wi-Fi at home, you are essentially connecting them to a LAN. A LAN, which can be wired or wireless, provides a high-speed connection allowing devices to communicate and share resources.

Wide Area Networks (WAN): A WAN spans a large geographical area, linking networks across cities, countries, or even continents. The Internet itself is a perfect embodiment of a WAN, connecting countless networks globally. WANs enable businesses with multiple locations to communicate and share resources seamlessly.

Metropolitan Area Networks (MAN): As the name suggests, MANs cover a larger area than LANs but are confined within the limits of a city or metropolitan area. They are often employed by Internet Service Providers (ISPs) to offer services to their clients in the same city. A city-wide Wi-Fi network is a typical example of a MAN.

Network Interface Cards (NIC): A NIC is a hardware component that enables a device to connect to a network. Whether wired or wireless, a NIC translates the data produced by the computer into a format suitable for network transmission.

Network Cables: Network cables, such as Ethernet cables or Fiber Optic cables, provide a physical medium for devices to communicate in a network. In contrast, wireless networking relies on radio waves, eliminating the need for physical cables.

8.PROGRAMMING

Programming is like giving instructions to a computer to solve a problem. Programming, also known as coding or software development, is the process of designing and building computer programs.

A program is a set of instructions that a computer follows to perform a specific task or solve a particular problem. These instructions are written in a programming language, which is a formal system of rules and symbols that is used to communicate with a computer.

The process of programming typically involves several key steps:

- Problem Definition: Understand the problem
- Algorithm Design: Make a plan (break it into steps)
- Coding: Write instructions in code
- Testing: Check instructions
- Debugging: Fix mistakes
- Documentation: Explain your code
- Maintenance: Keep improving



B

8.1.Creation of C#

C # is Microsoft's premier language for .NET development. It leverages time-tested features with cutting-edge innovations and provides a highly usable, efficient way to write programs for the modern enterprise computing environment. It is, by any measure, one of the most important languages of the 21st century.

The purpose of this chapter is to place C# into its historical context, including the forces that drove its creation, its design philosophy, and how it was influenced by other computer languages. This chapter also explains how C# relates to the .NET Framework. As you will see, C# and the .NET Framework work together to create a highly refined programming environment.

8.1.1.What is the .NET Framework?

The .NET Framework defines an environment that supports the development and execution of highly distributed, component-based applications. It enables differing computer languages to work together and provides for security, program portability, and a common programming model for the Windows platform. As

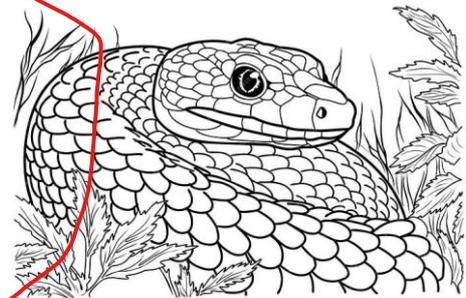
it relates to C#, the .NET Framework defines two very important entities. The first is the Common Language Runtime (CLR). This is the system that manages the execution of your program. Along with other benefits, the Common Language Runtime is the part of the .NET Framework that enables programs to be portable, supports mixed-language programming, and provides for secure execution.

The second entity is the .NET class library. This library gives your program access to the runtime environment. For example, if you want to perform I/O, such as displaying something on the screen, you will use the .NET class library to do it. If you are new to programming, then the term class may be new. Although it is explained in detail later in this book, for now a brief definition will suffice: a class is an object-oriented construct that helps organize programs. As long as your program restricts itself to the features defined by the .NET class library, your programs can run anywhere that the .NET runtime system is supported. Since C# automatically uses the .NET Framework class library, C# programs are automatically portable to all .NET environments.

8.2.PYTHON Programming Language

Prog

Python, renowned for its simplicity and readability, stands as one of the most versatile and widely adopted programming languages in the world today. Created by Guido van Rossum in the late 1980s, Python was designed with a focus on code readability, enabling developers to express concepts in fewer lines of code compared to languages like C++ or Java.

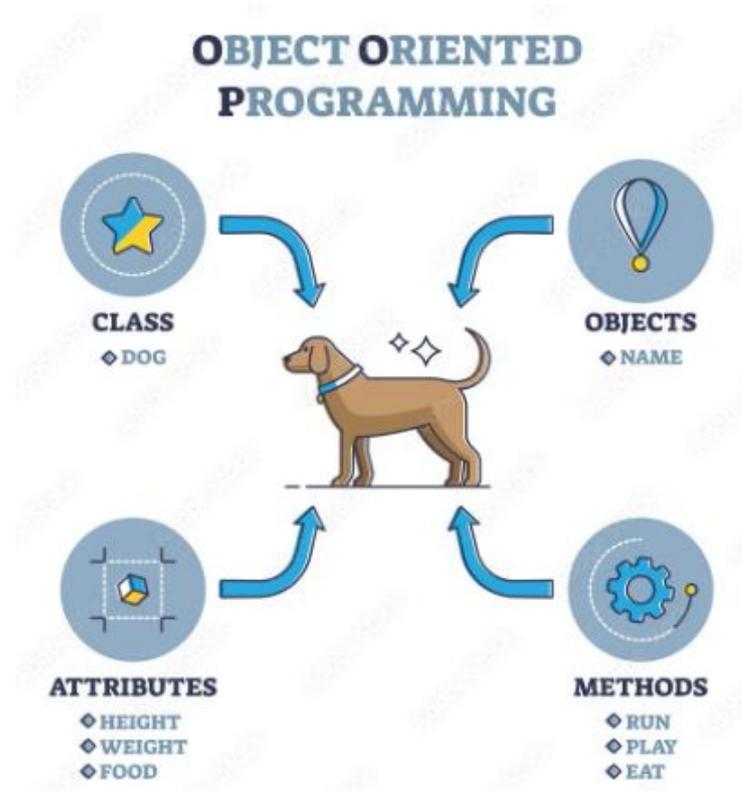


The Significance of Python

The importance of Python transcends traditional coding realms. Its versatility allows it to be employed in a multitude of applications, ranging from web development and scientific computing to data analysis, artificial intelligence, and more. Python's extensive library ecosystem empowers developers with pre-built modules and packages, easing the implementation of complex functionalities. This language's adaptability is showcased by its role in some of the most prominent technological advances of our time.

In web development, frameworks like Django and Flask have propelled Python to the forefront, enabling developers to build robust and scalable applications. In the realm of data science, Python, along with libraries like Pandas, NumPy, and Matplotlib, has become the de facto choice for data manipulation, analysis, and visualization. Additionally, Python's prowess in artificial intelligence and machine learning is exemplified by the popularity of libraries such as TensorFlow and PyTorch.

Beyond these domains, Python finds application in automation, scripting, game development, and more. Its straightforward syntax and vast community support make it an ideal choice for both novice programmers and seasoned developers alike.



8.3.Object Oriented Programing

Introduction to Object-Oriented Programming (OOP)

Object-Oriented Programming (OOP) stands as a fundamental paradigm in the world of software development, revolutionizing the way programmers approach problem-solving and software design.

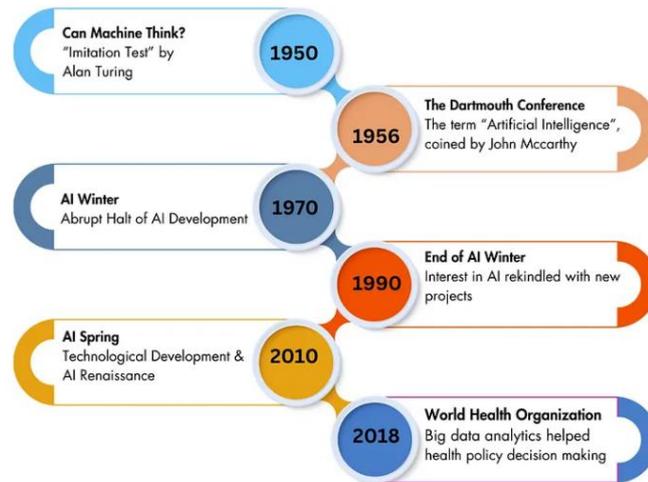
At its core, OOP introduces a novel way of structuring code by organizing data and its related functions into cohesive units called “objects.” This section will provide an insightful overview of OOP, highlighting its significance in modern software development and introducing key terminology that lays the foundation for a comprehensive understanding.

Definition and Concept of OOP

At its simplest, Object-Oriented Programming can be defined as a programming paradigm that models real-world entities and their interactions through the creation and manipulation of objects.

These objects are instances of classes, which act as blueprints or templates for creating objects. OOP promotes the idea of breaking down complex problems into manageable, modular components, making it easier to design, implement, and maintain software.

Timeline Diagram of Artificial Intelligence History

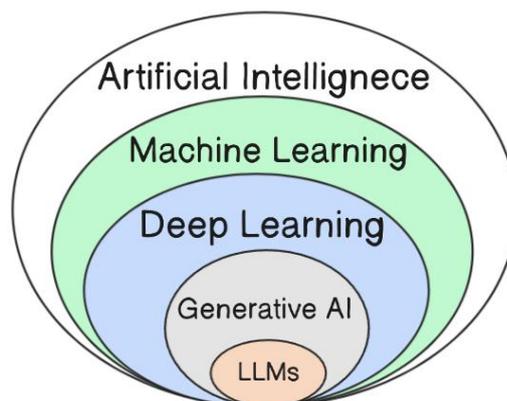


9. ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) has emerged as a transformative technology that is reshaping industries and revolutionizing the way we live and work. For business owners, managers, and curious beginners, understanding the fundamentals of AI is essential for making informed decisions and harnessing its potential to drive growth and innovation. In this article, we will delve deep into the world of AI, exploring its core principles, practical applications, and future implications.

9.1. Understanding Artificial Intelligence

Artificial Intelligence encompasses the development of computer systems capable of performing tasks that traditionally require human intelligence. The field includes various subfields, each with its own focus and application. Machine learning is one such subfield, where algorithms enable computers to learn from data and improve their performance over time. Natural language processing focuses on enabling machines to understand and interact with human language, while computer vision enables machines to interpret and understand visual information. Robotics combines AI with physical systems to create intelligent machines capable of interacting with the physical world.



Prog

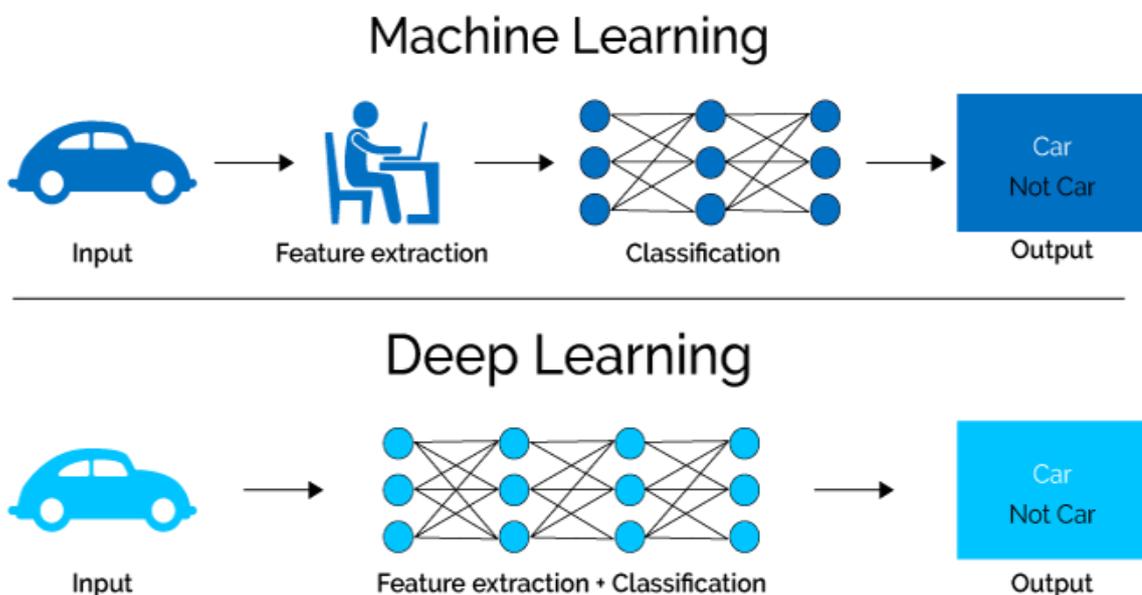
B 9.2. Machine Learning: The Essence of AI

Machine learning, a subset of AI, lies at the heart of many AI applications. It allows computers to learn from data and improve their performance over time without explicit programming. Machine learning algorithms identify patterns and relationships within data, extract meaningful insights, and make predictions or decisions based on those patterns. This ability to learn from experience enables machines to automate tasks, recognize speech, classify images, and more.

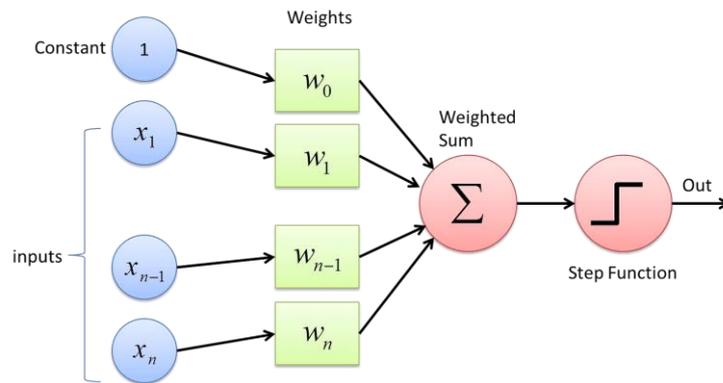
Machine learning is a fundamental subset of AI that plays a crucial role in enabling computers to learn from data and improve their performance over time. It provides machines with the ability to automatically learn patterns, make predictions, and make decisions without explicit programming.

In supervised learning, the algorithm learns from labeled examples to make predictions or classifications. It is provided with a dataset where each data point is associated with a target value or label. The algorithm analyzes the features of the data and learns to map them to the corresponding target value. This allows the algorithm to make accurate predictions or classifications for new, unseen data. For example, in a spam email classification system, the algorithm is trained on a dataset of labeled emails, where each email is classified as either spam or non-spam. By analyzing the features of the emails, such as the words used or the presence of certain keywords, the algorithm learns to distinguish between spam and non-spam emails. Once trained, it can accurately classify new emails as spam or non-spam.

Unsupervised learning, on the other hand, involves learning from unlabeled data, where the algorithm is not provided with any target labels. The algorithm analyzes the patterns and structures in the data to discover inherent relationships and groupings. Clustering algorithms are commonly used in unsupervised learning to group similar data points together. For example, in customer segmentation, an unsupervised learning algorithm can analyze customer data based on various attributes and group customers into segments based on their similarities.



9.3. Deep learning



9.3.1. Definition of Deep learning

Def: The term, Deep Learning, refers to training Neural Networks, sometimes very large Neural Networks.

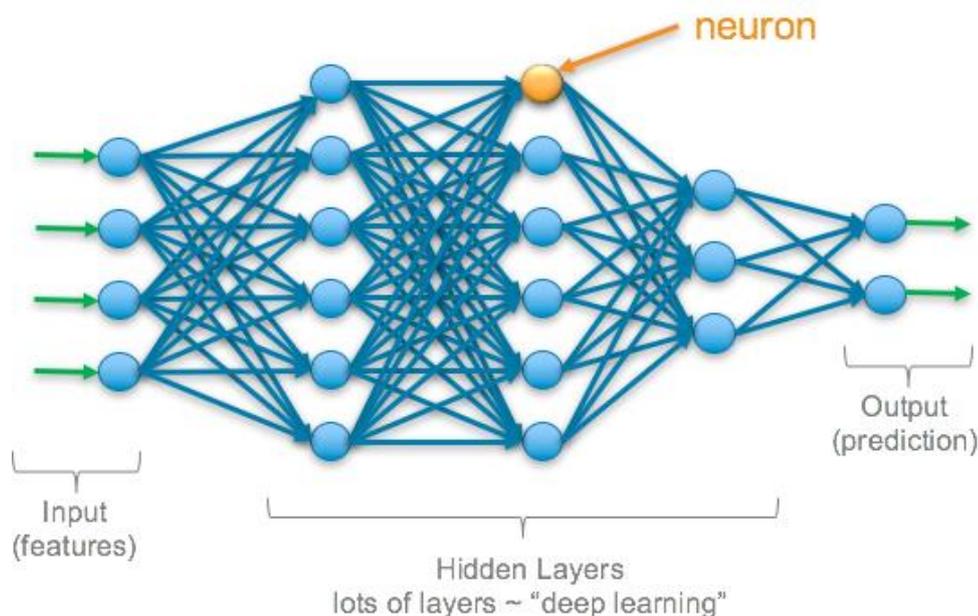
Def: Deep learning refers to the process of involving a system that thinks and learns exactly like humans using an artificial neural network

Def: Deep learning is a subset of machine learning that uses artificial neural networks to learn from data. the “deep” in deep learning refers to the depth of layers in a neural network.

Def: A neural network of more than three layers, including the inputs and the output, can be considered a deep-learning algorithm.

Def: Deep learning is a subset of machine learning based on artificial neural networks, with ‘deep’ referring to the multiple layers in these neural networks. While conventional machine learning algorithms are limited to relatively superficial analysis, deep learning models can extract richer, more abstract representations from large and complex datasets like images, text, and audio.

Application: Deep learning underpins today’s rapid advances in areas like computer vision, speech recognition, natural language processing, and artificial intelligence.



It is a subset of machine learning inspired by the human brain's structure and function. It uses artificial neural networks with multiple layers (hence the "deep") to process information. These networks learn and improve by analyzing massive amounts of data uncovering complex patterns and relationships. The "deep" in deep learning refers to the use of multiple layers, which allows the network to learn complex and hierarchical representations of data.

Deep learning models are trained on large amounts of data, such as images, text, or sound. During training, the network adjusts the connections between its artificial neurons to improve its ability to make predictions or classifications. There are different types of deep learning algorithms, each with its own strengths and weaknesses. Some common types include convolutional neural networks (CNNs) for image recognition, recurrent neural networks (RNNs) for language processing, and generative adversarial networks (GANs) for creating new data.

9.3.2. History of Deep learning

The first ANN was proposed in 1944, but it has become very popular in recent years. Deep learning was introduced in the early 50s but it became popular in recent years due to the increase in AI-oriented applications and the data that is being generated by the companies. While classical machine learning algorithms fell short of analyzing big data, artificial neural networks performed well on big data. The history of deep learning can be traced back to the early days of artificial intelligence (AI). In 1943, Warren McCulloch and Walter Pitts created a mathematical model of neurons in the brain, which was the first artificial neural network.

In the 1950s, Frank Rosenblatt developed the perceptron, a simple two-layer neural network that could be trained to recognize patterns. However, the perceptron had limitations, and it was not until the 1980s that neural networks began to be used more widely.

In the 1980s, Geoffrey Hinton and others developed a new type of neural network called the backpropagation algorithm. Backpropagation allowed neural networks to learn more complex patterns, and it led to a renewed interest in neural networks.

In the 1990s, deep learning research continued, but it was still a relatively niche field. However, in the early 2000s, there were a number of breakthroughs that led to a resurgence of interest in deep learning.

One of the most important breakthroughs was the development of the convolutional neural network (CNN). CNNs are a type of neural network that is specifically designed for image processing. They have been used to achieve state-of-the-art results in a variety of image recognition tasks, such as face recognition and object detection.

Another important breakthrough was the development of the recurrent neural network (RNN). RNNs are a type of neural network that is specifically designed for processing sequential data. They have been used to achieve state-of-the-art results in a variety of natural language processing tasks, such as machine translation and speech recognition.

In the past decade, deep learning has made significant progress in a wide variety of tasks, including image recognition, natural language processing, speech recognition, and machine translation. It is now one of the most active and promising areas of research in AI.

Prey

10.MICROSOFT WINDOWS

10.1.Power Shell

Pro

PowerShell is a versatile and powerful platform command-line shell and scripting language developed by Microsoft. It is designed for automating administrative tasks and configuration management on Windows, but it is now available on various platforms, including macOS and Linux. In this beginner's guide, we will cover the basics of PowerShell, what it is used for, and how to start using it.

PowerShell is a command-line shell that enables users to interact with their computer systems using text-based commands. It provides access to a wide range of system management and automation tasks. Unlike traditional command shells like Command Prompt, PowerShell uses a consistent and intuitive syntax based on verb-noun pairs, making it easier to learn and use.

10.2. What Can You Use PowerShell For?

PowerShell is a versatile tool with a wide range of applications, including:

Automation: PowerShell is excellent for automating repetitive tasks. You can write scripts to perform tasks such as file management, system configuration, and software installation.

System Administration: System administrators use PowerShell to manage and configure Windows servers and workstations. You can perform tasks like user management, network configuration, and system monitoring.

Scripting: PowerShell is a full-featured scripting language. You can write scripts to automate complex processes, making it a valuable tool for developers and IT professionals.

Reporting: PowerShell can retrieve and process data from various sources, including logs and databases, to generate reports and perform data analysis.

Cloud Computing: PowerShell integrates with cloud platforms like Microsoft Azure and AWS, allowing you to manage cloud resources and automate cloud-related tasks.

Active Directory Management: PowerShell is particularly powerful for managing Active Directory environments, making it essential for Windows domain administrators.

10.3.Windows Commands

COPY — Copies files to another location
 DIR — Displays files and folders in current directory
 DEL or ERASE — Deletes files
 Copy con — make a file via CMD
 Type (filename) — show a text file in cmd terminal
 EDIT — Starts file editor
 CD — Changes directory
 FIND — Finds a text string in the file
 MD or MAKEDIR — Creates a folder
 MOVE — Moves files from one folder to another
 RD or RMDIR — Deletes a folder
 REN or RENAME — Renames a file or folder

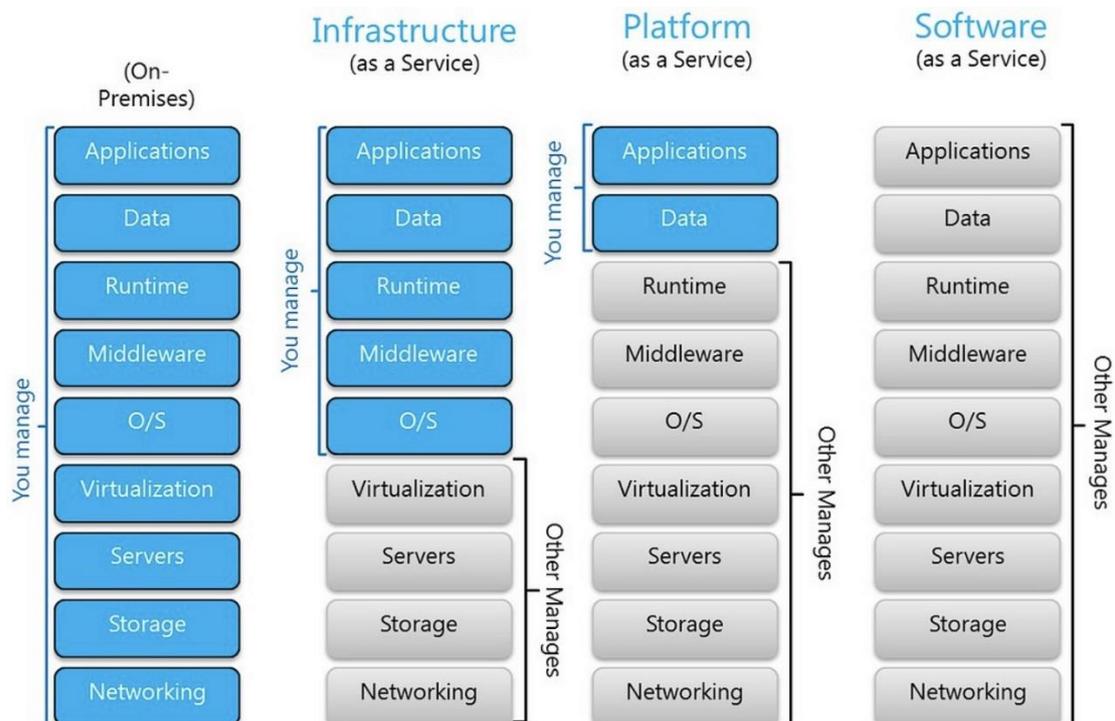
TREE — Shows directory structure of a disk or folder
 TYPE — Displays the contents of text files
 SHUTDOWN — Shutdowns or reboots your computer
 CHKDISK — Checks disk and shows statistics
 DEFRAG — Starts disk defragmentation
 DATE — Outputs or sets the current date
 TIME — Displays or sets the system time
 IPCONFIG — Shows information about network interfaces
 PING — Sends ICMP requests to the target host, checks host availability
 CLS — Clears screen
 CMD — Displays another command prompt
 HELP — Launches CMD help
 EXIT — Exits the command line

11. CLOUD COMPUTING

Prog

Cloud computing is the ability to access information and applications over the Internet. Cloud computing allows users to access applications and data from any location, as long as they have an Internet connection.

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.



11.1. Types of Cloud Computing Services

Infrastructure as a Service (IaaS)

Infrastructure as a service is a cloud computing model in which a third-party provider delivers computer infrastructure — typically a platform virtualization environment — as a service. Customers are able to access and use the provider's infrastructure, which typically includes operating systems, storage, and networking resources, over the Internet.

Platform as a Service (PaaS)

Platform as a service is a cloud computing model in which a third-party provider delivers a computing platform — typically a set of software tools — and infrastructure over the Internet. Customers are able to access and use the provider's platform to develop, run, and manage applications.

Software as a Service (SaaS)

Software as a service is a cloud computing model in which a third-party provider delivers software over the Internet. Customers are able to access and use the provider's software, typically through a web browser, while the provider manages the infrastructure and security.

12. Augmented Reality and Virtual Reality

Pr 9

The digital world has changed dramatically with the introduction of **Augmented Reality (AR)** and **Virtual Reality (VR)**. These exciting technologies blend the real and digital worlds, offering immersive experiences that transform how we see and interact with our surroundings.

AR enhances the real world by adding digital information to our physical environment. You may have seen it in popular apps like Pokémon Go or Snapchat filters, where digital elements appear in the real world, like virtual creatures or fun effects on your selfies.

VR, on the other hand, takes us to completely virtual environments. With special headsets, we can explore and interact with computer-generated worlds, feeling like we're truly there. It's like stepping into a different reality and experiencing things beyond our physical limits. These technologies go beyond gaming and have applications in education, healthcare, architecture, and more. They are revolutionizing how we learn, work, and interact with our surroundings.

12.1. Understanding Augmented Reality

Augmented Reality (AR) is a technology that overlays digital information, such as images, videos, or 3D models, onto the real world in real time. It enhances our perception of reality by seamlessly blending virtual elements with our physical environment. Unlike Virtual Reality, which creates an entirely simulated environment, AR enriches our existing surroundings, providing an interactive and immersive experience.

The core principles of AR revolve around real-time tracking, registration, and rendering. Real-time tracking involves accurately tracking the position and orientation of the user or device in the environment. Registration refers to aligning the virtual content with real-world objects, ensuring that the virtual elements appear to be in the right place. Rendering focuses on seamlessly integrating virtual content into the real world, considering lighting, shadows, and other environmental factors to make it look more natural.

One of the key strengths of AR is its ability to enhance the real-world environment by adding contextual information and interactive elements. For example, imagine wearing AR glasses and walking through a city. As you look around, information about nearby landmarks, restaurants, or historical facts is displayed in your field of view. You can receive real-time directions, reviews, or ratings, all overlaid onto the physical world. AR can provide users with a deeper and more engaging experience by supplementing their perception with relevant digital information.

12.2. Exploring Virtual Reality

Users can explore and interact with a computer-generated, virtual environment thanks to a technology called virtual reality (VR). It immerses individuals in a completely artificial world, isolating them from the physical environment and stimulating their senses to provide a sense of presence and realism. With specialized hardware and software, VR transports users to virtual realms, enabling unique and immersive experiences.

The basic principles of VR involve creating a convincing illusion of presence and interactivity. Presence refers to the feeling of being physically present in the virtual environment, despite knowing that it is not real. It is achieved through a combination of sensory inputs, including visual, auditory, and sometimes haptic (touch) feedback. By delivering a seamless and convincing experience, VR tricks the brain into perceiving the virtual environment as real.

To experience VR, users typically rely on hardware and equipment designed specifically for this purpose. Headsets are the primary component, worn on the head to provide a visual and auditory experience. These headsets often include high-resolution displays that cover the user's field of view, delivering stereoscopic 3D visuals that create a sense of depth and immersion. They may also incorporate built-in headphones or speakers for spatial audio, enhancing the audio experience.

VR systems frequently feature controllers or input devices that allow users to interact with the virtual environment in addition to headsets. These controllers can track the user's hand movements, enabling them to manipulate objects, perform gestures, and navigate within the virtual space. Some advanced VR setups may include full-body tracking systems, haptic feedback devices, or motion platforms that further enhance the sense of immersion and realism.

12.3. Differentiating between AR and VR technologies:

AR and VR differ primarily in their level of immersion and the way they interact with the real world:

1. Augmented Reality (AR): AR superimposes digital data on the physical world, improving our experience of reality. Users can see and interact with virtual elements that appear to coexist with their physical environment. AR does not replace the real world but supplements it with contextual information.

2. Virtual Reality (VR): VR, on the other hand, creates a completely simulated environment that isolates users from the physical world. By wearing a VR headset, users are immersed in a virtual world that blocks out their surroundings. The real world is replaced with a computer-generated environment in virtual reality.

Pro