# 1. Data Representation

## 1.1 Number Systems

### Understanding Binary

Why binary? Computers use binary (base 2) because electronic components have two states: on (1) and off (0). This makes it efficient and reliable for processing data.
Number Systems:

- Denary: Base 10, uses digits 0-9.
- Binary: Base 2, uses digits 0 and 1.
- Hexadecimal: Base 16, uses digits 0-9 and letters A-F.

**Conversions:**

- **Denary to Binary:** Divide by 2 repeatedly, noting remainders. Read remainders from bottom to top.
    - Example: Convert 25 to binary.
        - 25 / 2 = 12 remainder 1
        - 12 / 2 = 6 remainder 0
        - 6 / 2 = 3 remainder 0
        - 3 / 2 = 1 remainder 1
        - 1 / 2 = 0 remainder 1
        - Binary equivalent: 11001
- **Denary to Hexadecimal:** Divide by 16 repeatedly, noting remainders. Convert remainders greater than 9 to their hexadecimal equivalent.
    - Example: Convert 257 to hexadecimal.

        - 257 / 16 = 16 remainder 1
        - 16 / 16 = 1 remainder 0
        - Hexadecimal equivalent: 101
- **Hexadecimal to Binary:** Convert each hexadecimal digit to its 4-bit binary equivalent.
    - Example: Convert A5 to binary.
        - A = 1010, 5 = 0101
        - Binary equivalent: 10100101

Hexadecimal: Used for representing memory addresses, colors, and machine code efficiently due to its compact form. Binary Addition: Add corresponding bits, carry over if sum is 2. Overflow: Occurs when the result of a binary addition exceeds the capacity of the register (e.g., 8 bits).

**Binary Shifts and Two's Complement**

Binary Shifts:

- Left shift: Multiplies the number by 2.
- Right shift: Divide the number by 2. Bits shifted out are lost, zeros are shifted in.

Example: Left shift of 1101 (13) by 1 position: 11010 (26) Example: Right shift of 1101 (13) by 1 position: 0110 (6)

Two's Complement: Used to represent negative numbers in binary.

- Invert all bits of the positive number.
- Add 1 to the result.

**Note:** Binary shifts are efficient for multiplication and division by powers of 2

### 1.2 Text, Sound, and Images

#### Text Representation

- **Character Sets:** Convert text characters into binary code for computer processing.
    - **ASCII:** 7-bit code for English characters.
    - **Unicode:** Larger character set, supports multiple languages and symbols.
- **Comparison:** Unicode requires more bits per character than ASCII for increased representation.

#### Sound Representation

- **Sampling:** Sound waves are converted into digital form by taking samples at regular intervals.
    - **Sample rate:** Number of samples taken per second, affects sound quality and file size.
    - **Sample resolution:** Number of bits used to represent each sample, affects dynamic range and file size.

#### Image Representation

- **Pixels:** Images are composed of pixels, each represented by binary data.
    - **Resolution:** Number of pixels in an image, affects image detail and file size.
    - **Color depth:** Number of bits used to represent each pixel's color, affects color accuracy and file size.

### 1.3 Data Storage and Compression

**Data Storage Units**

- **Units:** Bit, nibble, byte, kilobyte (KB), megabyte (MB), gigabyte (GB), and larger units.
- **Calculations:** Determine file size based on given information (e.g., image resolution, color depth, sound sample rate, resolution, and duration).

**Data Compression**

1. **Purpose:** Reduce file size for efficient storage and transmission.
2. **Methods:**
   a. **Lossless:** Reduces file size without data loss (e.g., RLE).
   b. **Lossy:** Reduces file size by permanently removing data (e.g., reducing image resolution or color depth, reducing sound sample rate or resolution).

# 2. Data Representation

## 2.1 Types and Methods of Data Transmission

Data is transmitted across networks in discrete units called **packets**. Each packet contains a **header** (with sender/receiver addresses, sequence number, error checks), the actual data (**payload**), and a **trailer** for error detection.

**Packet switching** is a method where data is broken into packets, which can travel independently through a network. This offers flexibility and efficiency but can lead to packet loss or reordering.

**Data transmission modes** include:

- **Simplex:** One-way communication (e.g., radio broadcast).
- **Half-duplex:** Communication in both directions but not simultaneously (e.g., walkie-talkie).
- **Full-duplex:** Simultaneous communication in both directions (e.g., telephone).

Data can be sent **serially** (bit by bit) or **parallel** (multiple bits simultaneously). **USB** is a high-speed serial interface for connecting peripherals to computers.

### 2.2 Methods of Error Detection

To ensure data integrity during transmission, **error detection** techniques are used:

- **Parity check:** An extra bit is added to indicate if the number of 1s in a data unit is odd or even. Simple but only detects single-bit errors.
- **Checksum:** A mathematical calculation is performed on the data, and the result is appended. More complex than parity checks.
- **Echo check:** The data is sent back to the sender for comparison, but it's inefficient.
- **Check digit:** An extra digit is added based on an algorithm (e.g., ISBN, UPC).
- **Automatic Repeat Request (ARQ):** The receiver sends acknowledgments, and if errors occur, the data is retransmitted.

### 2.3 Encryption

**Encryption** safeguards data confidentiality by converting it into an unreadable format.

- **Symmetric encryption** uses a single secret key for both encryption and decryption.

- **Asymmetric encryption** employs a public key for encryption and a private key for decryption, offering higher security but with increased computational demands.

# 3. Hardware

## 3.1 Computer Architecture

**The CPU is the "brain" of a computer, executing instructions and processing data. A microprocessor is a single-chip CPU.**

**Key CPU components include:**

- **Arithmetic Logic Unit (ALU): Performs calculations and logical operations (like comparing numbers).**
- **Control Unit (CU): Manages the overall operation of the CPU, coordinating the fetch-decode-execute cycle.**
- **Registers: High-speed storage areas within the CPU for temporary data and instructions.**
- **Buses: Data pathways connecting different components within and outside the CPU.**

**The CPU follows the fetch-decode-execute cycle to process instructions:**

1. **Fetch: Retrieves the next instruction from memory.**
2. **Decode: Interprets the instruction to determine the required operation.**
3. **Execute: Performs the specified operation using the ALU and other components.**

**CPU performance is influenced by factors like the number of cores (multiple processing units), cache size (fast memory for storing frequently used data), and clock speed (how fast the CPU operates). The instruction set is the vocabulary of commands a CPU understands.**

**Embedded systems are specialized computers designed for specific tasks (e.g., microwaves, cars) with limited functionality compared to general-purpose computers.**

### 3.2 Input and Output Devices

**Input devices capture data from the outside world (e.g., keyboard, mouse, camera). Output devices display or produce information (e.g., monitor, printer, speakers). Sensors convert physical stimuli (like light, temperature, or motion) into electrical signals for computers to process.**

### 3.3 Data Storage

**Primary storage (RAM and ROM) is directly accessible by the CPU. RAM (Random Access Memory) is temporary and volatile (loses data when power is off). ROM (Read Only Memory) is non-volatile and stores essential system instructions.**

**Secondary storage is non-volatile and used for long-term data storage. Types include:**

- **Magnetic storage (hard drives) uses magnetic properties to store data.**
- **Optical storage (CDs, DVDs) uses lasers to read and write data.**
- **Solid-state storage (SSDs, flash drives) uses electronic circuits for fast and durable storage.**

**Virtual memory is a technique that allows the OS to use hard drive space as if it were additional RAM. Cloud storage stores data on remote servers accessible via the internet.**

### 3.4 Network Hardware

**A network interface card (NIC) enables a computer to connect to a network. Each device on a network has a unique MAC address (Media Access Control). An IP address (Internet Protocol) is a numerical label assigned to devices for communication. Routers direct data packets between networks.**

**There are two main versions of IP addresses: IPv4 and IPv6, with IPv6 being designed to accommodate a larger number of devices.**

# 4. Software

## 4.1 Types of Software and Interrupts

**Software** is the intangible part of a computer system, instructing the hardware on what to do. It's divided into two main types:

- **System Software:** This underpins the computer's operation, managing hardware and providing essential services.
    - **Operating Systems (OS):** The core software that controls hardware, runs applications, and provides a user interface.
      It handles tasks like file management, memory allocation, and input/output operations.
    - **Utility Software:** Performs specific maintenance or management tasks (e.g., antivirus software, disk defragmenters).
- **Application Software:** Designed for specific user tasks, such as word processing, gaming, or web browsing.

**How Software Interacts:** The hardware is the foundation, providing the physical components. Firmware, embedded software within hardware, acts as an intermediary. The operating system then sits on top, managing both hardware and software.

**Interrupts** are signals that temporarily pause a program's execution to handle urgent tasks. They can be triggered by hardware (e.g., keyboard press, mouse movement) or software (e.g., arithmetic errors, system calls). An **Interrupt Service Routine (ISR)** is a dedicated piece of code that handles the interrupt and then returns control to the original program.

### 4.2 Types of Programming Language, Translators, and IDEs

**Programming Languages** are used to create software. They're categorized into:

- **High-Level Languages:** Human-readable and easier to write (e.g., Python, Java, C++).
- **Low-Level Languages:** Closer to machine code, offering better performance but harder to understand (e.g., assembly language).

**Translators** convert code into machine-readable format:

- **Compilers:** Translate entire code into executable form before running.
- **Interpreters:** Translate code line by line during execution.

**Integrated Development Environments (IDEs)** provide a comprehensive platform for software development, including code editing, debugging, and compilation tools. They streamline the development process.

## 5. The Internet and Its Use

### 5.1 The Internet and the World Wide Web

The **Internet** is a global network of interconnected computers. It provides the infrastructure for communication and data transfer. The **World Wide Web (WWW)** is a system of interconnected documents (web pages) accessible through the internet.

A **URL** (Uniform Resource Locator) is the address of a webpage. It typically includes the protocol (HTTP or HTTPS), domain name, and file path. **HTTP** (Hypertext Transfer Protocol) is the protocol used for transferring web pages. **HTTPS** is a secure version of HTTP that encrypts data.

**Web browsers** are software applications that allow users to access and interact with the web. They interpret HTML (Hypertext Markup Language), the language used to create web pages. When you enter a URL, the browser:

1. **Resolves the domain name:** Uses a DNS (Domain Name System) to convert the URL into an IP address.

2. **Establishes a connection:** Connects to the web server hosting the webpage.
3. **Requests the page:** Sends an HTTP request to the server.
4. **Receives the page:** The server sends the HTML code back to the browser.
5. **Renders the page:** The browser interprets the HTML and displays the webpage.

**Cookies** are small text files stored on a user's device. They can be used to track user preferences, maintain shopping cart items, and store login information.

## 5.2 Digital Currency

A **digital currency** is a virtual or electronic currency that operates independently of a central bank or government. Unlike traditional currencies, it exists solely in electronic form.

**Blockchain** is a decentralized digital ledger that records transactions across multiple computers. It provides a secure and transparent way to track digital currency transactions.

## 5.3 Cyber Security

**Cybersecurity** is the practice of protecting computer systems and networks from digital attacks. Common threats include:

- **Hacking:** Unauthorized access to computer systems.
- **Malware:** Malicious software (viruses, worms, Trojans, spyware, adware, ransomware).
- **Phishing:** Deceiving users into revealing personal information.
- **Pharming:** Redirecting users to fake websites.
- **Denial-of-service (DoS) attacks:** Overloading a system to prevent legitimate users from accessing it.
- **Brute-force attacks:** Trying every possible combination of characters to guess a password.
- **Data interception:** Unauthorized access to data transmitted over networks.

To protect against these threats, various security measures can be implemented:

- **Anti-malware software:** Detects and removes malware.
- **Firewalls:** Act as a barrier between a network and the internet.
- **Strong passwords and authentication:** Use complex passwords and multi-factor authentication.
- **Software updates:** Keep software up-to-date with the latest security patches.
- **Privacy settings:** Control the information shared online.
- **Secure connections (HTTPS):** Encrypt data transmitted over the internet.
- **Backups:** Regularly create copies of important data.
- **User education:** Raise awareness about cyber threats and prevention.

## 6. Automated and Emerging Technologies

### 6.1 Automated Systems

**Automated systems** combine **sensors**, **microprocessors**, and **actuators** to perform tasks with minimal human intervention. **Sensors** gather data from the environment, **microprocessors** process this data, and **actuators** carry out actions based on the processed information.

**Advantages** of automation include increased efficiency, accuracy, and productivity. **Disadvantages** include initial setup costs, potential job displacement, and reliance on technology. Automation is prevalent in industries like manufacturing, transportation (e.g., self-driving cars), agriculture (e.g., drones for crop monitoring), and weather forecasting.

### 6.2 Robotics

**Robotics** is the field of creating and using robots. A **robot** typically has a mechanical structure, electrical components (sensors, microprocessors, actuators), and programmable control systems. Robots are employed in various sectors, including manufacturing, healthcare (e.g., surgical robots), exploration (e.g., Mars rovers), and domestic tasks (e.g., vacuum cleaners). While offering precision, endurance, and the ability to handle hazardous tasks, robots can be expensive, require maintenance, and may lead to job displacement.

### 6.3 Artificial Intelligence (AI)

**AI** aims to create intelligent agents, systems that can reason, learn, and adapt. Key components of AI systems include a **knowledge base** (stored information), a **rule base** (decision-making rules), and an **inference engine** (to draw conclusions).

**Expert systems** are a type of AI that applies knowledge to solve problems within a specific domain. **Machine learning** enables systems to learn from data without explicit programming, improving performance over time.

# New version prt 2

**arning** enable
Here is the structured and uniform representation of your content:

---

# Data Representation

# 1.1 Number Systems

## Understanding Binary

**Why binary?**
Computers use binary (base 2) because electronic components have two states: on (1) and off (0). This makes it efficient and reliable for processing data.

**Number Systems:**

- **Denary:** Base 10, uses digits 0-9.
- **Binary:** Base 2, uses digits 0 and 1.
- **Hexadecimal:** Base 16, uses digits 0-9 and letters A-F.

**Conversions:**

**Denary to Binary:**
Divide by 2 repeatedly, noting remainders. Read remainders from bottom to top.

- **Example:** Convert 25 to binary.
    1. 25 / 2 = 12 remainder 1
    2. 12 / 2 = 6 remainder 0
    3. 6 / 2 = 3 remainder 0
    4. 3 / 2 = 1 remainder 1
    5. 1 / 2 = 0 remainder 1
       Binary equivalent: 11001

**Denary to Hexadecimal:**
Divide by 16 repeatedly, noting remainders. Convert remainders greater than 9 to their hexadecimal equivalent.

- **Example:** Convert 257 to hexadecimal.
    1. 257 / 16 = 16 remainder 1
    2. 16 / 16 = 1 remainder 0
       Hexadecimal equivalent: 101

**Hexadecimal to Binary:**
Convert each hexadecimal digit to its 4-bit binary equivalent.

- **Example:** Convert A5 to binary.
    - A = 1010
    - 5 = 0101
      Binary equivalent: 10100101

**Hexadecimal:**
Used for representing memory addresses, colors, and machine code efficiently due to its compact form.

**Binary Addition:**
Add corresponding bits, carry over if sum is 2.

- **Overflow:** Occurs when the result of a binary addition exceeds the capacity of the register (e.g., 8 bits).

### Binary Shifts and Two's Complement

**Binary Shifts:**

- **Left shift:** Multiplies the number by 2.
- **Right shift:** Divides the number by 2. Bits shifted out are lost, zeros are shifted in.
    - **Example:** Left shift of 1101 (13) by 1 position: 11010 (26)
    - **Example:** Right shift of 1101 (13) by 1 position: 0110 (6)

**Two's Complement:**
Used to represent negative numbers in binary.

1. Invert all bits of the positive number.
2. Add 1 to the result.

*Note: Binary shifts are efficient for multiplication and division by powers of 2*

---

## 1.2 Text, Sound, and Images

### Text Representation

**Character Sets:**
Convert text characters into binary code for computer processing.

- **ASCII:** 7-bit code for English characters.
- **Unicode:** Larger character set, supports multiple languages and symbols.

**Comparison:**
Unicode requires more bits per character than ASCII for increased representation.

### Sound Representation

**Sampling:**
Sound waves are converted into digital form by taking samples at regular intervals.

- **Sample rate:** Number of samples taken per second, affects sound quality and file size.
- **Sample resolution:** Number of bits used to represent each sample, affects dynamic range and file size.

### Image Representation

**Pixels:**
Images are composed of pixels, each represented by binary data.

- **Resolution:** Number of pixels in an image, affects image detail and file size.
- **Color depth:** Number of bits used to represent each pixel's color, affects color accuracy and file size.

---

## 1.3 Data Storage and Compression

### Data Storage Units

**Units:**
Bit, nibble, byte, kilobyte (KB), megabyte (MB), gigabyte (GB), and larger units.

**Calculations:**
Determine file size based on given information (e.g., image resolution, color depth, sound sample rate, resolution, and duration).

### Data Compression

**Purpose:**
Reduce file size for efficient storage and transmission.

**Methods:**

- **Lossless:** Reduces file size without data loss (e.g., RLE).
- **Lossy:** Reduces file size by permanently removing data (e.g., reducing image resolution or color depth, reducing sound sample rate or resolution).

---

# Data Transmission

# 2.1 Types and Methods of Data Transmission

**Data is transmitted across networks in discrete units called packets.**
Each packet contains a header (with sender/receiver addresses, sequence number, error checks), the actual data (payload), and a trailer for error detection.

**Packet switching:**
Data is broken into packets, which can travel independently through a network. This offers flexibility and efficiency but can lead to packet loss or reordering.

**Data transmission modes include:**

- **Simplex:** One-way communication (e.g., radio broadcast).
- **Half-duplex:** Communication in both directions but not simultaneously (e.g., walkie-talkie).
- **Full-duplex:** Simultaneous communication in both directions (e.g., telephone).

**Data can be sent serially (bit by bit) or parallel (multiple bits simultaneously).**
USB is a high-speed serial interface for connecting peripherals to computers.

# 2.2 Methods of Error Detection

**To ensure data integrity during transmission, error detection techniques are used:**

- **Parity check:** An extra bit is added to indicate if the number of 1s in a data unit is odd or even. Simple but only detects single-bit errors.
- **Checksum:** A mathematical calculation is performed on the data, and the result is appended. More complex than parity checks.
- **Echo check:** The data is sent back to the sender for comparison, but it's inefficient.
- **Check digit:** An extra digit is added based on an algorithm (e.g., ISBN, UPC).
- **Automatic Repeat Request (ARQ):** The receiver sends acknowledgments, and if errors occur, the data is retransmitted.

# 2.3 Encryption

**Encryption safeguards data confidentiality by converting it into an unreadable format.**

- **Symmetric encryption:** Uses a single secret key for both encryption and decryption.
- **Asymmetric encryption:** Employs a public key for encryption and a private key for decryption, offering higher security but with increased computational demands.

# Hardware

## 3.1 Computer Architecture

**The CPU is the "brain" of a computer, executing instructions and processing data.**
A microprocessor is a single-chip CPU.

**Key CPU components include:**

- **Arithmetic Logic Unit (ALU):** Performs calculations and logical operations (like comparing numbers).
- **Control Unit (CU):** Manages the overall operation of the CPU, coordinating the fetch-decode-execute cycle.
- **Registers:** High-speed storage areas within the CPU for temporary data and instructions.
- **Buses:** Data pathways connecting different components within and outside the CPU.

**The CPU follows the fetch-decode-execute cycle to process instructions:**

1. **Fetch:** Retrieves the next instruction from memory.
2. **Decode:** Interprets the instruction to determine the required operation.
3. **Execute:** Performs the specified operation using the ALU and other components.

**CPU performance is influenced by factors like the number of cores (multiple processing units), cache size (fast memory for storing frequently used data), and clock speed (how fast the CPU operates).**
The instruction set is the vocabulary of commands a CPU understands.

**Embedded systems:**
Specialized computers designed for specific tasks (e.g., microwaves, cars) with limited functionality compared to general-purpose computers.

## 3.2 Input and Output Devices

**Input devices** capture data from the outside world (e.g., keyboard, mouse, camera).
**Output devices** display or produce information (e.g., monitor, printer, speakers).
**Sensors** convert physical stimuli (like light, temperature, or motion) into electrical signals for computers to process.

## 3.3 Data Storage

**Primary storage (RAM and ROM):**
Directly accessible by the CPU. RAM (Random Access Memory) is temporary and volatile (loses data when power is off). ROM (Read Only Memory) is non-volatile and stores essential system instructions.

**Secondary storage:**
Non-volatile and used for long-term data storage. Types include:

- **Magnetic storage (hard drives):** Uses magnetic properties to store data.
- **Optical storage (CDs, DVDs):** Uses lasers to read and write data.
- **Solid-state storage (SSDs, flash drives):** Uses electronic circuits for fast and durable storage.

**Virtual memory:**
A technique that allows the OS to use hard drive space as if it were additional RAM.
**Cloud storage:**
Stores data on remote servers accessible via the internet.

## 3.4 Network Hardware

**Network interface card (NIC):**
Enables a computer to connect to a network. Each device on a network has a unique MAC address (Media Access Control).
**IP address (Internet Protocol):**
A numerical label assigned to devices for communication.
**Routers:**
Direct data packets between networks.

**There are two main versions of IP addresses:**

- **IPv4**
- **IPv6:** Designed to accommodate a larger number of devices.

---

# Software

## 4.1 Types of Software and Interrupts

**Software is the intangible part of a computer system, instructing the hardware on what to do.**
It's divided into two main types:

- **System Software:** Manages hardware and provides a platform for running application software.
- **Application Software:** Performs specific user tasks (e.g., word processors, games).

**System software includes:**

- **Operating Systems (OS):** Manages computer resources and provides a user interface (UI).
- **Utility Software:** Maintains and optimizes computer performance (e.g., antivirus, disk cleanup).

**Interrupts:**
Signals that indicate an event needing immediate attention, pausing the current process to execute the interrupt handler.

## 4.2 Programming Languages

**Programming languages enable humans to write instructions for computers.**
There are three main types:

- **Machine Language:** Binary code understood directly by the CPU.
- **Assembly Language:** Low-level, uses mnemonics and requires an assembler.
- **High-level Languages:** More abstract, easier to read and write (e.g., Python, Java, C++), and require compilers or interpreters.

**Source code:**
Human-readable instructions written in a high-level language.
**Object code:**
Machine-readable instructions produced by a compiler.

**Interpreters and Compilers:**

- **Interpreters:** Translate and execute code line-by-line, making debugging easier.
- **Compilers:** Translate the entire code into machine language before execution, producing faster runtime performance.

---

## 4.3 Operating Systems

**The OS is the most important system software, managing all other software and hardware.**
Functions include:

- **File Management:** Organizes data into files and folders.

- **Process Management:** Manages running programs.
- **Memory Management:** Allocates and tracks memory usage.
- **Device Management:** Controls hardware devices.
- **User Interface (UI):** Provides a way for users to interact with the computer.

**Types of user interfaces:**

- **Graphical User Interface (GUI):** Uses windows, icons, and menus.
- **Command-Line Interface (CLI):** Uses text commands.

**Virtual Memory:** An extension of physical RAM, uses part of the hard drive to simulate additional RAM.

---

## 4.4 Databases

**Databases are structured collections of data, managed by Database Management Systems (DBMS).**
They support various operations:

- **CRUD:** Create, Read, Update, Delete.

**Relational Databases:**
Organize data into tables with rows and columns, using SQL (Structured Query Language) for management.

**Data Types:**
Specify the kind of data (e.g., integer, string, date).

**Normalization:**
A process to organize data to reduce redundancy and improve integrity.

---

# The Internet and Its Use

## 5.1 The Internet and the World Wide Web

**The Internet** is a global network of interconnected computers providing the infrastructure for communication and data transfer. **The World Wide Web (WWW)** is a system of interconnected documents (web pages) accessible through the Internet.

**URL (Uniform Resource Locator):**

- The address of a webpage.
- Typically includes the protocol (HTTP or HTTPS), domain name, and file path.

**HTTP (Hypertext Transfer Protocol):**
Used for transferring web pages.
**HTTPS:**
A secure version of HTTP that encrypts data.

**Web Browsers:**
Software applications that allow users to access and interact with the web. They interpret HTML (Hypertext Markup Language), the language used to create web pages.

**How Browsers Work:**

1. **Resolves the domain name:** Uses DNS (Domain Name System) to convert the URL into an IP address.
2. **Establishes a connection:** Connects to the web server hosting the webpage.
3. **Requests the page:** Sends an HTTP request to the server.
4. **Receives the page:** The server sends the HTML code back to the browser.
5. **Renders the page:** The browser interprets the HTML and displays the webpage.

**Cookies:**
Small text files stored on a user's device. Used to track user preferences, maintain shopping cart items, and store login information.

## 5.2 Digital Currency

**Digital Currency:**
Virtual or electronic currency that operates independently of a central bank or government. Exists solely in electronic form.

**Blockchain:**
A decentralized digital ledger that records transactions across multiple computers. Provides a secure and transparent way to track digital currency transactions.

## 5.3 Cyber Security

**Cybersecurity:**
The practice of protecting computer systems and networks from digital attacks.

**Common Threats:**

- **Hacking:** Unauthorized access to computer systems.

- **Malware:** Malicious software (viruses, worms, Trojans, spyware, adware, ransomware).
- **Phishing:** Deceiving users into revealing personal information.
- **Pharming:** Redirecting users to fake websites.
- **Denial-of-service (DoS) attacks:** Overloading a system to prevent legitimate users from accessing it.
- **Brute-force attacks:** Trying every possible combination of characters to guess a password.
- **Data interception:** Unauthorized access to data transmitted over networks.

**Security Measures:**

- **Anti-malware software:** Detects and removes malware.
- **Firewalls:** Act as a barrier between a network and the Internet.
- **Strong passwords and authentication:** Use complex passwords and multi-factor authentication.
- **Software updates:** Keep software up-to-date with the latest security patches.
- **Privacy settings:** Control the information shared online.
- **Secure connections (HTTPS):** Encrypt data transmitted over the Internet.
- **Backups:** Regularly create copies of important data.
- **User education:** Raise awareness about cyber threats and prevention.

---

# Automated and Emerging Technologies

## 6.1 Automated Systems

**Automated Systems:**
Combine sensors, microprocessors, and actuators to perform tasks with minimal human intervention.

- **Sensors:** Gather data from the environment.
- **Microprocessors:** Process this data.
- **Actuators:** Carry out actions based on the processed information.

**Advantages:**

- Increased efficiency.
- Improved accuracy.
- Enhanced productivity.

**Disadvantages:**

- Initial setup costs.
- Potential job displacement.
- Reliance on technology.

**Applications:**

- Manufacturing.
- Transportation (e.g., self-driving cars).
- Agriculture (e.g., drones for crop monitoring).
- Weather forecasting.

## 6.2 Robotics

**Robotics:**
The field of creating and using robots.

- **Mechanical structure.**
- **Electrical components:** Sensors, microprocessors, actuators.
- **Programmable control systems.**

**Applications:**

- Manufacturing.
- Healthcare (e.g., surgical robots).
- Exploration (e.g., Mars rovers).
- Domestic tasks (e.g., vacuum cleaners).

**Advantages:**

- Precision.
- Endurance.
- Ability to handle hazardous tasks.

**Disadvantages:**

- High cost.
- Maintenance requirements.
- Potential job displacement.|

## 6.3 Artificial Intelligence (AI)

**Artificial Intelligence (AI):**
Aims to create intelligent agents, systems that can reason, learn, and adapt.
Key components of AI systems include:

- **Knowledge base:** Stored information.
- **Rule base:** Decision-making rules.
- **Inference engine:** To draw conclusions.

**Expert Systems:**
A type of AI that applies knowledge to solve problems within a specific domain.

**Machine Learning:**
Enables systems to learn from data without explicit programming, improving performance over time.