Computer Science Lecture - Part 1: Programming Foundations

Programming involves writing instructions for computers using languages like Python, Java, and C++. Programs follow syntax and logic to perform tasks. Variables store data; data types classify values (integers, floats, strings). Control structures like if-statements and loops enable decision-making and repetition. Functions encapsulate reusable code blocks. Debugging corrects logical and syntax errors.

Computer Science Lecture - Part 2: Data Structures

Data structures organize data efficiently. Arrays store elements in contiguous memory. Linked lists use nodes pointing to next elements. Stacks (LIFO) and queues (FIFO) manage ordered data access. Trees represent hierarchical relationships; binary search trees enable fast lookup. Graphs model networks with nodes and edges. Choosing structures depends on operations and performance.

Computer Science Lecture - Part 3: Algorithms

Algorithms solve problems step-by-step. Sorting algorithms (bubble, insertion, merge, quicksort) arrange data. Searching algorithms (linear, binary search) find values. Algorithm analysis measures efficiency using Big O notation. Recursive algorithms solve problems by reducing to subproblems. Dynamic programming optimizes overlapping subproblems. Algorithm design balances correctness, efficiency, and simplicity.

Computer Science Lecture - Part 4: Object-Oriented Programming

Object-oriented programming (OOP) models real-world entities as objects with attributes and behaviors. Classes define blueprints; objects are instances. Principles include encapsulation (hiding data), inheritance (reusing parent class features), polymorphism (overriding methods), and abstraction (simplifying complexity). OOP promotes modular, maintainable, scalable code. Languages supporting OOP include Java, C++, Python.

Computer Science Lecture - Part 5: Databases

Databases store structured data for retrieval and manipulation. Relational databases use tables linked by keys; SQL queries access data (SELECT, INSERT, UPDATE, DELETE). Normalization reduces redundancy. Transactions ensure consistency. NoSQL databases handle unstructured or scalable data. Database management systems (DBMS) like MySQL, PostgreSQL, MongoDB support various applications from web to enterprise systems.

Computer Science Lecture - Part 6: Networking Basics

Networking enables communication between devices. Key concepts include IP addresses, protocols (TCP/IP, HTTP, FTP), DNS translating domain names, and client-server architecture. Packets transfer data over routers and switches. Firewalls secure networks. Wireless networking uses standards like Wi-Fi. The Internet interconnects global networks, supporting email, web, streaming, and cloud services.

Computer Science Lecture - Part 7: Cybersecurity

Cybersecurity protects systems from attacks. Threats include malware, phishing, denial-of-service, SQL injection. Defenses include encryption, authentication, firewalls, intrusion detection. Cybersecurity policies define access controls and incident response. Ethical hacking tests vulnerabilities. Data privacy laws regulate protection of personal information. Cybersecurity evolves with technology and attacker tactics.

Computer Science Lecture - Part 8: Software Development Lifecycle

The software development lifecycle (SDLC) guides project stages: requirements gathering, design, implementation, testing, deployment, maintenance. Models include waterfall (sequential) and agile (iterative). Version control (e.g., Git) tracks changes. Testing (unit, integration, system, user acceptance) ensures quality. Documentation facilitates maintenance. SDLC balances time, cost, scope, and quality.

Computer Science Lecture - Part 9: Artificial Intelligence

Artificial Intelligence (AI) enables machines to perform tasks requiring human intelligence: learning, reasoning, perception, decision-making. Machine learning algorithms train models on data to predict outcomes. Deep learning uses neural networks for image and speech recognition. Natural language processing allows computers to understand text. AI applications span healthcare, finance, transportation, education, and entertainment.