Final Exam Topics (CS 2026)
Informatics

1. PROCESSORS AND MEMORIES

The von Neumann Principles and the von Neumann Cycle. Processor architecture and functions, superscalar architecture. The role of instructions and registers (overflow, underflow, and numerical representation precision). Data, Address, and Control Buses. Types of Memories (readable/writeable) and classification criteria (physical structure, organization, access type, speed).

Task: Demonstrate runtime and memory optimization techniques in a chosen programming language.

2. SECONDARY STORAGE, PERIPHERALS, AND I/O

Classification of secondary storage based on architecture, perating principles, primary parameters, and comparative analysis. Peripherals and their classification (based on function, communication direction, and mode). Fundamentals of I/O management, programmed I/O, IRQ and DMA controlled I/O.

Task: File access and handling in a chosen programming language.

3. OPERATING SYSTEMS

Definition and tasks of an operating system. Process management and scheduling. Scheduling methods and algorithms in interactive systems. Requirements and goals of scheduling. Memory management, relocation, memory protection, swapping, and paging.

Task: Searching in log files within a chosen OS (PowerShell, Bash, etc.).

4. COMPUTER NETWORK ARCHITECTURE

The role and basic operation of network devices (network interface, repeater, bridge, hub, switch, router). The characteristics and operation of Ethernet. IEEE 802.11 network standards. The Internet Protocol. Structure of IP addresses, problems related to IP addresses and possible solutions (NAT, CIDR), IPv6. Routing, management of routing tables. TCP and UDP. QoS parameters.

Task: Draw a typical computer network for a family home or micro-enterprise. Mark the different types of connections, devices, and possible IP addresses that can be used for communication.

5. NETWORK PROTOCOLS

The structure and operation of DNS. Comparison of REST and SOAP-based communication. Network protocols used in software development: HTTP, HTTPS, SSH, FTP, SMTP. HTTP response (status) codes (2xx-5xx, the most common error codes), HTTP methods, the role of HTTP headers (Authorization, Set-Cookie, Accept, Content-Type, Content-Disposition, Location, User-Agent).

Task: Designing a REST API containing GET, POST, PUT, DELETE requests, based on an application description.

6. COMPUTER NETWORK SECURITY

Encrypted and unencrypted communication. Classic encryption, symmetric and public key algorithms. Key sharing problems, Diffie-Hellman key exchange. Cracking codes with different methods (social engineering, analytical attack, brute-force attack). External and internal networks, the role and operation of firewalls, DMZ. Security issues, vulnerabilities, risks and risk reduction in wireless networks.

Task: What solution would you recommend to a software development company if they need to discuss the steps and results of a development project with their industrial partner in an encrypted form? Consider as many aspects as possible and justify your answer.

7. SQL

General characteristics of SQL. Classification (DDL, DML, DCL) and syntax. Automation (triggers, stored procedures, functions, and constraints, etc.).

Task: Demonstrate grouping and aggregation capabilities through your unique example (SELECT, WHERE, GROUP BY, ORDER BY, JOIN, aggregation).

8. DATABASE DESIGN

Advantages and disadvantages of Hierarchical, Relational, and NoSQL data models. Relational database principles, the relational design process, and the goals of normalization (1NF-5NF). Fundamental relational database design patterns. Data Warehouse design: star schema, snowflake schema, multidimensional models, and On-line Analytical Processing.

Task: Design a relational database with 4-5 tables for a unique, self-provided example.

9. ORM (Object-Relational Mapping)

Concepts: ORM, persistence, and entities. Software and database connection: Representation of the database within the software in a chosen programming language.

Task: Demonstrate establishing a database connection and representing the database in a chosen programming language.

10. ALGORITHMS AND DATA STRUCTURES

Definition, specification, and description of algorithms. Algorithm efficiency. Data structures and their applications (sequential and associative structures, lists, collections, FIFO, LIFO, binary heaps, hash tables, and search trees). Tree operations (search, insertion, deletion, and balancing). Sorting algorithms (insertion, merge, quick, heap, counting, and radix. Graphs and graph algorithms: BFS and DFS searches, traversals (pre-, in-, and postorder), spanning trees (Prim's and Kruskal's algorithms), and shortest path algorithms (Dijkstra and Bellman-Ford).

Task: Demonstrate one of the above algorithms using a unique example and run it on a sample input.

11. IMPERATIVE LANGUAGES

Types, arithmetic, relational, and logical operations. Control structures (conditional and unconditional control flow, loops). Functions, procedures (subroutines) parameter passing, local and global variables. The role of the stack in function calls.

Task: Introduce a chosen programming language (compilation – compiler/interpreter, execution, packaging, frameworks, virtual machines).

12. OBJECT-ORIENTED PROGRAMMING (OOP)

Core concepts of object-oriented programing: object, class, method, inheritance, polymorphism, and encapsulation.

Task: In a chosen object-oriented programming language, define a class that models the operation of simple household appliances (e.g. a hairdryer, toaster, etc.), including attributes, a constructor, and getter/setter methods (Imagine if the device were a 'smart appliance', what kind of user interface would you design for it?). Focus on the efficient modeling of similar devices using inheritance.

13. SYSTEM DESIGN AND SOFTWARE DEVELOPMENT

UML: use case, sequence, and entity-relationship diagrams. Version Control: Git basics (branch, pull, push, commit, tag, merge) and their practical benefits. Agile methodologies (e.g., SCRUM). Prototyping models. Software documentation.

Task: Present and explain unique examples for the aforementioned UML diagrams.

Final Exam Topics (CS 2026)
Mathematics

1. SET THEORY BASICS

Sets, operations on sets. Relations, functions. Injective, onto, and bijective functions. Equivalence relations. Ordering relations. Natural, integer, rational, real, and complex numbers: their properties, operations and ordering properties.

2. LOGICS

Propositional logics: propositions (statements), operations with propositions, formulas, formalization, disjunctive and conjunctive normal forms, inferences, inference rules. Predicate logic: predicates, quantifiers, formulae, formalization and interpretation, inferences in predicate logics.

3. LINEAR ALGEBRA

Real vector spaces, normed spaces. Vector operations. Inner (scalar) product. Linear independence, basis, dimension. Matrices and linear operators. Homogeneous and inhomogeneous linear systems of equations. The determinant and trace of matrices. Eigenvalues and eigenvectors, spectral decomposition of symmetric matrices.

4. CALCULUS

Convergence of sequences and series. Taylor-series. Limits and continuity of functions. Differential calculus of one- and multivariable functions. Finding function minima and maxima. Convexity of functions. Real integral calculus, definite and indefinite integrals.

5. PROBABILITY THEORY

Discrete and continuous random variables. Probability distributions and probability densities. Independence of random variables. Joint probabilities, marginal distributions, expectation value, variance, correlation. Laws of large numbers. Central limit theorem.

6. NUMBER THEORY

Greatest common divisor. Euclidean division. Euclidean algorithm. Primes and non-factorizable numbers. Unique prime factorization. Systems of linear Diophantine equations, linear congruences. Euler’s theorem, Chinese remainder theorem. Number theoretic functions. Multiplicativity. Sum and inverse functions.

7. OPERATION RESEARCH

Linear programming and solution with simplex algorithm. Integer programming and mixed integer programming. Network Flow Algorithms, Optimization on Networks. Duality and Duality theorems (Weak Duality, Strong Duality and Complementary theorem).

8. NUMERICAL METHODS FOR SOLVING EQUATIONS AND EQUATION SYSTEMS

Direct methods for solving Lienar Equation Systems (Gaussian elimination, solution with LU-decomposition, solution with QR-decomposition) Iterative methods for solving Linear Equation Systems (Fixed point theorem, conditions for the contraction property, classical iterations) Numerical methods of non-linear equations (bisection method, fixed point iteration, Newton-Raphson method).

9. INTERPOLATION AND QUADRATURES

The basic problem of the interpolation. Methods of polinomial interpolation (Lagrange and Newton form of the interpolation polinomial) Least squares method (the basic problem and the LES for the solution) Interpolation type quadratures. Gauss-type and Chebishev-quadratures. Classical quadratures.