Bachelor State Final Exam Topics

The aim of these pages is to disseminate a brief overview of Bachelor State Final Examination topics in order to clarify (1) what specific knowledge belongs to each general topic, (2) how deep a knowledge will be required, and (3) what subjects are covered.

This version has been prepared for the Bachelor State Final Exams starting September 2019.

Mathematics

1. Numbers 2. Fundamentals of Differential Calculus 3. Integration 4. Fundamentals of the theory of functions of several variables 5. Metric spaces 6. Basic algebraic structures 7. Vector spaces 8. Scalar product 9. Solving systems of linear equations 10. Matrices 11. Determinants 12. Eigenvalues and eigenvectors 13. Discrete mathematics 14. Graph theory 15. Probability and statistics 16. Logic 17. OI Algebra

Computer science

1. Automata and languages 2. Algorithms and data structures 3. Databases 4. Programming languages 5. Computer Architecture and Operating Systems 6. OI Networking fundamentals 6. PSS+SDI Networking fundamentals 7. OI Optimization methods 7. PSS+SDI Compilers 8. OI AO Algorithms and optimization 8. OI DMS Discrete models and structures 8. OI ML Computational linguistics 8. PSS PG Computer graphics 8. PSS SP System programming 8. PSS SIT Networks and Internet Technologies 8. SDI DW Databases and web 8. SDI SI Software Engineering

1. Numbers

Properties of natural numbers, integers, rationals, reals and complex numbers. Sequences and series. Cauchy sequences.

- Properties of natural numbers, integers and rational numbers (algebraic), and real numbers (completeness, uncountability).
- Sequences and series of real numbers and their properties (properties of limits and series summation, convergence criteria).

Covered by lectures

- NMAI054 Mathematical Analysis I

2. Fundamentals of Differential Calculus

Real functions of one variable. Continuity, limit of a function at a point (both finite and infinite). Specific functions (polynomials, rational functions, trigonometric and inverse trigonometric functions, logarithms and exponential functions). Derivatives: definitions and basic rules, intermediate value theorem, higher order derivatives. Applications (function of a function, Taylor polynomial with remainder).

- Real functions of one variable.
- Continuity, limit of a function at a point (both finite and infinite).
- Specific functions (polynomials, rational functions, trigonometric and inverse trigonometric functions, logarithms and exponential functions).
- Derivatives: definitions and basic rules, intermediate value theorem, higher order derivatives.
- Applications (function of a function, Taylor polynomial with remainder).

Covered by lectures

- NMAI054 Mathematical Analysis I
- NMAI055 Mathematical Analysis II

3. Integration

Antiderivatives, methods of calculation. Definite (Riemann) integral, applications of definite integrals. Multiple integrals and Fubini's theorem.

- Antiderivatives (primitive functions): definition, uniqueness, existence, methods of calculation.
- Riemann and Newton integral (definitions, properties, existence, relationship) and their applications (areas and volumes, estimating sum of a series).

Covered by lectures

- NMAI054 Mathematical Analysis I
- NMAI055 Mathematical Analysis II

4. Fundamentals of the theory of functions of several variables

Partial derivatives and total derivative, intermediate value theorems, extremes of multivariate functions, implicit function theorem.

- Partial derivative and total derivative (definition, existence, basic rules, relationship between them).
- Extremes of functions of several variables (existence of extrema, criteria for local extrema using partial derivatives, method of Lagrange multipliers).
- Implicit function theorem.

Covered by lectures

- NMAI055 Mathematical Analysis II

5. Metric spaces

Definition of metric space, examples. Continuity, open and closed sets. Compactness.

- Definition of metric space, examples (Euclidean spaces, function spaces, non-Archimedean spaces).
- Open and closed sets (closure properties).
- Continuous maps between spaces (including topological definition).
- Compactness, especially in Euclidean spaces.

Covered by lectures

- NMAI055 Mathematical Analysis II

6. Basic algebraic structures

Group, rings, fields - definitions and examples. Little Fermat theorem. Divisibility and decomposition of polynomials into irreducible factors. Decomposition into root factors for polynomials with real, rational, complex coefficients. Multiplicity of roots and relationship to polynomial derivatives.

- Groups, subgroups
  - definition
  - examples
  - commutativity
- Algebraic fields
  - definition
  - characteristic
  - finite and infinite fields
- Polynomials over fields

Canceled

- Cyclic groups
- Fundamentals of universal algebra
- Boolean algebras and lattices
- Rings and ideals

Covered by lectures

- NMAI062 Algebra I

7. Vector spaces

Groups, fields. Basic properties of vector spaces, subspaces, span, linear dependence and independence. Steinitz Exchange Lemma. Finally generated vector spaces, bases. Linear transformations.

- Vector spaces and subspaces, their properties, basic notions (linear combination, linear span (hull), generators, linear dependence and independence, basis, dimension, coordinates) and their application.
- Steinitz Exchange Lemma with the idea of proof.
- How to test for linear dependence and independence, find a basis, determine subspace dimension, etc.
- Matrix spaces (row, column and null space).
- Linear transformations and representation by matrices, one-to-one linear transformations and their equivalent characterization, composition of linear transformations, finding kernel and image.
- Isomorphism, properties and relationship to vector space dimension.

Covered by lectures

- NMAI057 Linear Algebra I

8. Scalar product

Properties in real and complex case. Norms. Cauchy-Schwarz inequality. Orthogonality. Orthogonal complement and its properties.

- Scalar product and its properties.

– Norm and relationship with scalar product, examples.
– Cauchy-Schwarz inequality, including proof for real case.
– Orthogonality, orthonormal basis, properties and applications (e.g. finding coordinates and projection).
– Gram-Schmidt orthogonalization, idea of proof and practical skill in using it.
– Orthogonal projection, formula and geometrical applications.
– Orthogonal complement and properties, relation to orthonormnal basis and null space of matrix, practical skill in finding orthogonal complement.
– Orthogonal matrices, their properties and relationship with orthonormal bases and linear transformations.

Covered by lectures

– NMAI058 Linear Algebra II

9. Solving systems of linear equations

Linear subspaces of a vector space, geometric interpretation. Solution of system of linear equations is a subspace. Rank condition for solvability of inhomogeneous system of linear equations (Frobenius theorem). Solution of a system of equations by matrix operations. Connection between solution of system of equations and orthogonal complement.

– Systems of linear equations and solution set.
– Gauss and Gaussian-Jordan elimination methods, reduced row echelon form, uniqueness (without proof).
– Matrix rank and its relationship to solvability, Frobenius theorem and its interpretation.
– Description of solution set and relationship to affine spaces.
– Null space of a matrix and applications (to properties of linear transformations, finding eigenvectors etc.).

Covered by lectures

– NMAI057 Linear Algebra I
– NMAI058 Linear Algebra II (linear transformations)

10. Matrices

Matrices and their rank. Operations on matrices and their properties. Inverse matrix. Invertible matrices and their various characterizations. Matrices and linear transformations, change of basis.

– Matrices and operations on matrices (sum, product, transposition, etc.), interpretation of product of matrices as composition of linear transformations.
– Rank of a matrix and its transpose.
– Special matrices: symmetric, diagonal, triangular, invertible (non-singular), orthogonal, positive definite and semi-definite.
– Invertible matrices and equivalent characterizations (from the point of view of systems of equations, rank, determinant, eigenvalues, linear transformations), basic properties and examples (elementary matrix products, etc.).
– Inverse of a matrix: existence, properties, calculation and use (e.g. to solve systems of equations).
– Positive definite and positive semi-definite matrices: properties and characterization by decomposition and by eigenvalues.
– Methods for testing positive definiteness, Cholesky decomposition, and Sylvester's criterion, including ability to apply to examples.

Covered by lectures

– NMAI057 Linear Algebra I

– NMAI058 Linear Algebra II

11. Determinants

Definitions and basic properties of the determinant. Modification of determinants, calculation. Geometric meaning of the determinant. Minors and inverse matrices. Cramer's rule.

– Determinants and their properties (e.g. multiplication of determinants, linearity, relationship with invertibility and matrix eigenvalues).
– Knowledge of how to calculate the determinant, modification of determinants, Laplace expansion.
– Geometric meaning of the determinant.
– Cramer's rule.

Covered by lectures

– NMAI058 Linear Algebra II

12. Eigenvalues and eigenvectors

Eigenvalues and eigenvectors of linear transformations and square matrices. Their calculation and basic properties. Diagonalization of a matrix with distinct eigenvalues. Jordan normal form of a matrix in general.

– Eigenvalues and eigenvectors, their geometric meaning and properties, multiple eigenvalues, spectral radius.
– Characteristic polynomial, relation of eigenvalues to roots of this polynomial (both directions).
– Calculation of eigenvalues and eigenvectors.
– Similarity, diagonalizability (necessary and sufficient conditions) and basic knowledge of Jordan normal form.
– Eigenvalues and eigenvectors of symmetric matrices, spectral decomposition.

Covered by lectures

– NMAI058 Linear Algebra II

13. Discrete Mathematics

Ordered sets. Set systems, matchings, matching in bipartite graphs (system of distinct representatives). Combinatorial counting. Principle of inclusion and exclusion.

– Relations, properties of binary relations (reflexivity, symmetry, antisymmetry, transitivity).
– Equivalence relations and partition into equivalence classes.
– Partial orders, basic terms (minimal and maximal elements, minimum element and maximum element, chain, antichain), height and width of a partially ordered set, theorem relating height and width to size.
– Functions, types of functions (injective, surjective, bijective), and the number of functions between two finite sets of a given type.
– Permutations and their basic properties (their number, fixed points, etc.).
– Binomial coefficients and relationships between them, binomial theorem and its applications.
– Estimates of factorials and binomial coefficients.
– Principle of inclusion and exclusion, general formulation (and proof), applications (hat-check problem, Euler's totient function, number of surjections, etc.).

- Set systems and hypergraphs.
- Hall's theorem on systems of distinct representatives, relation to matching in bipartite graphs, principle of its proof and algorithmic aspects (polynomial-time algorithm for finding SDR).
- Finite projective planes (FPP), axiomatic definition, order of FPP (well-defined) and relationship of order to number of points, finite projective plane duality, existence of FPP of given order.
- Ordinary generating functions, basic properties and principles of enumeration with them. Generalized binomial theorem (without proof, only used for calculation).
- Applications of generating functions to combinatorial counting and recurrences (Fibonacci and Catalan numbers).

Covered by lectures

- NDMI002 Discrete Mathematics
- NDMI011 Combinatorics and Graph Theory I

14. Graph Theory

Basic concepts of graph theory, graph representation. Trees and their basic properties, spanning trees. Eulerian graphs. Plane graphs, graph colouring. Menger's theorem, cut-flow duality.

- Basic concepts (graphs, vertices and edges, graph isomorphism, subgraph, vertex neighbourhood and degree, graph complement, bipartite graph), basic examples of graphs (complete graph and complete bipartite graph, paths and cycles).
- Number of graphs and number of non-isomorphic graphs on a given set of vertices.
- Graph connectivity, connected components, distance between vertices.
- Characterization of Eulerian graphs (principle of proof), variant for directed graphs.
- Trees, definitions and basic properties (existence of leaves, number of edges in a tree), equivalent characterizations of trees.
- Spanning trees, definitions and basic properties, minimal spanning tree problem. Number of spanning trees of complete graphs (Cayley formula, without proof) and methods for enumeration of number of spanning trees of general graphs.
- Plane graphs, definitions and fundamental concepts (planar graph and plane drawing of a graph, faces), Euler's formula and maximum number of edges in a planar graph (proof and applications).
- Chromatic number, definition of proper colouring, relationship between chromatic number and clique number, d-degenerate graphs and estimation of their chromatic number. Chromatic number of planar graphs (principle of proof of five colour theorem).
- Edge- and vertex-connectivity of graphs, cuts and characterization using disjoint paths (edge and vertex version of Menger's theorem, applications rather than proof).
- Ear decomposition of 2-connected graphs and applications.
- Fundamentals of Ramsey theory, Ramsey's theorem for graphs (for 2 colours).
- Directed graphs, strong and weak connectivity.
- Network flows. Definition of network and of flow in it, existence of maximum flow (without proof), principle of searching for maximum network flow with integer-valued capacities (e.g. using Ford-Fulkerson algorithm).

Covered by lectures

- NDMI002 Discrete Mathematics
- NDMI011 Combinatorics and Graph Theory I

15. Probability and statistics

Random events, conditional probability, independence of random events. Random variables, mean value, distribution of random variables, normal and binomial distribution. Linear combinations of random variables. Point estimates, confidence intervals, hypothesis testing, t-test, chi-square test, linear regression.

- Random events, conditional probability, independence of random events
  - definition of these terms
  - Bayes' formula
  - applications
- Random variables, mean value, distribution of random variables, geometric, binomial and normal distribution.
- Linear combination of random variables
  - linearity of expectation
  - applications
- Point estimates, confidence intervals, hypothesis testing.

Cancelled topics

- t-test
- chi-squared test
- linear regression

Covered by lectures

- NDMI002 Discrete Mathematics
- NMAI059 Probability and Statistics

16. Logic

Language, formulas, semantics, tautology. Decidability, satisfiability, truth, provability. Theorem on compactness and completeness of propositional and predicate logic. Normal form of formulas in propositional logic, prenex normal form of formulas in predicate logic.

- Syntax
  - knowledge of and working with basic syntax of propositional and predicate logic (language, open and closed formulas, etc.)
- Normal forms of propositional formulas, prenex normal form of predicate logic formulas
  - knowledge of basic normal forms (CNF, DNF, PNF)
  - converting to normal form
  - applications in algorithms (SAT, resolution)
- Semantics
  - truth, falsity, independence of a formula with respect to a theory, satisfiability, tautology, consequence
  - concepts of model theory, extension of theories

- Analysis of propositional theories over finite languages
- Provability
  - the concept of formal proof
  - ability to work with the various formal proof systems (e.g. tableau method, resolution, Hilbert-style calculus)
- Compactness and completeness theorems for propositional and predicate logic
  - formulation and understanding of importance
  - proof idea
  - examples of applications
- Decidability
  - the concept of completeness and its criteria
  - relevance to decidability
  - examples of decidable theories

17. OI Algebra

Subgroup, normal subgroup, factor group, ideal. Homomorphisms of groups and other structures. Field of fractions.

- Divisibility in commutative cancellative monoids.
- Principal ideals and Euclidean domains. Polynomial rings, multiplicity of roots, evaluation homomorphism, cyclicity of finite multiplicative subgroups of fields.
- Rupture field and splitting field of a polynomial, decomposition of polynomials.
- Finite fields. Finding irreducible polynomials over finite fields.
- Free algebras, terms and varieties.

Covered by lectures

- NMAI063 Algebra II

1. Automata and languages

Chomsky hierarchy, classes of automata and grammars, determinism and non-determinism. Closure properties of classes of languages.

- Regular languages
  - finite automata, the language accepted by a finite automaton, deterministic, non-deterministic, epsilon transitions
  - regular expressions, Kleene's theorem
  - pumping lemma for finite automata
  - regular grammars
- Context-free languages
  - context-free grammars, language generated by a grammar, Chomsky normal form of a grammar
  - pushdown automata, class of languages accepted by pushdown automata
  - deterministic pushdown automata, acceptance by final state and by empty stack, relationship between sets of languages accepted
  - pumping lemma for context-free languages
- Context-sensitive languages
  - context-sensitive and noncontracting (monotonic) grammars, linear-bounded automata
- Turing machines
  - Turing machines, type-0 grammars, diagonal language, universal language

---

- Chomsky hierarchy
  - determine the equivalence or inclusion of classes of languages generated by automata and grammars mentioned above
  - classify a particular language in the Chomsky hierarchy, construct a corresponding automaton or grammar and use pumping lemmas to prove that it is not in a more basic class

---

- Closure properties
  - (non)closure of regular, context-free and deterministic context-free languages under union, intersection, complement, homomorphism and inverse homomorphism

Covered by lectures

- NTIN071 Automata and grammars

2. Algorithms and data structures

Time complexity of algorithms, worst-case and average case complexity. Complexity classes P and NP, reducibility, NP-completeness. “Divide and conquer” technique - applications and complexity analysis, dynamic programming. Binary search trees, balancing, heaps. Hashing. Sequential sorting, comparison based algorithms, bucket sorting, sorting networks. Graph algorithms - depth-first search and breadth-first search, connected components, strongly connected components of a directed graph, topological sorting, shortest paths, spanning trees, network flows. Transitive closure. Text search algorithms. Algebraic algorithms - DFT, Euclid’s algorithm. RSA. Approximation algorithms.

- Time complexity of algorithms
  - definition of the RAM computational model
  - time and space computation for a particular input
  - time and space complexity of an algorithm
  - data size measurement
  - difference between complexity in the best, worst and average case
  - asymptotic notation: O, Ω, Θ
- Complexity classes
  - P and NP
  - problem transformations and reductions
  - NP-hard and NP-complete problems
  - examples of NP-complete problems and reductions between them
- “Divide and conquer”
  - principle of recursive division of a problem into subproblems
  - calculating complexity using recurrence equations
  - Master Theorem
  - applications: merge sort, multiplication of large numbers, Strassen’s algorithm
- Dynamic programming
  - principle of dynamic programming (subproblem solutions from smallest to largest)
  - applications: longest increasing subsequence, edit distance
- Binary search trees
  - definition of a search tree
  - operations with unbalanced trees
  - AVL trees and their balancing
  - red-black trees and their balancing
- Heaps
  - Binary heaps
- Hashing
  - hashing with buckets
  - open addressing
  - analysis of average time complexity
- Sorting
  - primitive sorting algorithms (bubble sort, insertion sort, etc.)
  - heapsort
  - quicksort, randomized pivot selection, average complexity analysis
  - estimate of lower bound for complexity of comparison sorting algorithms
  - bucket sorting of numbers and strings
  - parallel sorting using comparator networks
- Graph algorithms
  - breadth-first and depth-first search
  - detection of connected components
  - topological sort of a directed graph
  - transitive closure
  - strongly connected components of directed graphs
  - shortest paths in weighted graphs: Dijkstra’s algorithm and Bellman–Ford algorithm
  - minimum spanning tree: Jarník and Borůvka algorithms
  - network flows: Ford–Fulkerson, Dinic and Goldberg algorithms
- Text search algorithms
  - Knuth–Morris–Pratt and Aho–Corasick algorithms

- Algebraic algorithms
  - Euclid's algorithm
  - discrete Fourier transform and its applications
  - computation of Fourier transform with FFT algorithm
- RSA
  - encryption, decryption and key generation
- Approximation algorithms
  - absolute and relative error
  - approximation schemes
  - examples: travelling salesman, knapsack

Covered by lectures

- NTIN060 Algorithms and Data Structures I
- NTIN061 Algorithms and Data Structures II
- NPRG030 Programming I
- NPRG031 Programming II
- NMAI062 Algebra I

3. Databases

Database system architectures. Conceptual, logical and physical database design. Algorithms for design of relational schemas, normal forms, referential integrity. Transaction processing, transaction properties, locking protocols, deadlocks. ER diagrams, methods for IS design. Overview of SQL.

- Database system architectures
- Conceptual, logical and physical views of data
  - Conceptual modeling - ER, UML (data diagrams)
  - Logical data model - primarily the relational data model
  - Physical data model - files (heaps, indexed sequential files, sorted files)
- Algorithms for design of relational schemas, normal forms, referential integrity
  - Reasons to normalize relations
  - First, second and third normal forms and BCNF
  - Decomposition and synthesis of relations
  - Primary and foreign keys
- Transaction processing, transaction properties, locking protocols, deadlocks
  - ACID properties of transactions
  - Scheduling transactions
  - Serial and serializable schedules
  - Conflict-equivalent schedules
  - Conflict serializability, testing with a precedence graph
  - Two-phase locking protocol
  - Strict two-phase locking protocol
  - Resolving the deadlock problem in a database
  - SQL transaction isolation
- ER diagrams, methods for IS design
  - Entities, relations, attributes and their representation in a conceptual model
  - Translating a conceptual model into a (relational) logical data model
  - Cardinality and arity of relations and their representation in the relational model
- Overview of SQL
  - Composing basic commands
    - Basic queries (SELECT - FROM - WHERE)
    - Sorting (ORDER BY)
    - Grouping data and aggregation (GROUP BY - HAVING, MIN, MAX, AVG, COUNT)
    - Joining tables (INNER JOIN, OUTER JOIN0
    - Subqueries, operators IN, EXISTS, ALL, ANY
    - Testing for NULL values (IS NULL, IS NOT NULL)

    - Understanding and explaining basic functionality
        - Stored procedures
        - Functions
        - Triggers

Covered by lectures

- NDBI025 Database systems
- NDBI026 Database applications

4. Programming languages

Implementation principles of object-oriented languages, runtime support. Separate compilation, linking, build management. Concepts and principles of object-oriented design. Generic programming, templates and generics, compile-time polymorphism. Non-procedural programming.

- Concepts and principles of object-oriented design
    - classes, interfaces, methods, attributes, inheritance
        - visibility of members
        - namespaces
        - separation into packages/modules
    - multiple inheritance and its problems
        - language-specific methods for resolving problems
        - multiple and virtual inheritance in C++
        - single inheritance and default methods in Java
    - implementing interfaces
    - polymorphism
        - static vs. dynamic polymorphism
    - functional elements of object-oriented languages
        - function objects, lambdas, support in standard libraries
    - important design patterns (singleton, factory …)
- implementation of object-oriented languages
    - basic object-oriented concepts in a concrete language (Java, C++, C#)
    - primitive type vs. objects
        - implementation of primitive types
        - memory representation of compound types and objects
    - implementation of virtual methods (virtual method tables)
    - lifetime of objects
        - allocating and initializing objects (statically, on the stack, on the heap)
            - constructors, calling inherited constructors
        - freeing objects
            - explicit delete/dispose
            - garbage collection
            - automatically freeing objects, shared_ptr/unique_ptr
            - destructors, finalizers
    - collection types
        - standard collection types, their attributes and usage
        - implementing custom containers
    - threads and synchronization
        - implementing threads
        - basic synchronization constructs
        - data types with atomic operations
    - debugging, exceptions
        - throwing and catching exceptions (try-catch-finally)
        - working with resources (try-with-resources (Java), RAII (C++), using (C#))
    - reflection and introspection (Java, C#)
        - basic usage

    - type inference (C++)
        - auto, decltype, type inference of template parameters
- Generic programming, templates and generics, compile-time polymorphism
    - how generic types (templates) work in a given language
    - covariance and contravariance of generics
    - policy classes, traits, problems of operations with unknown types
- Non-procedural programming
    - logic programming (Prolog)
        - basic constructs (predicates, lists, cut)
        - unification
    - functional programming
        - functional support in object-oriented languages
- Separate compilation, linking, compiler directives
    - compilation vs. interpreting
    - role of linking
    - JIT

The concrete formulation of questions will correspond to the field of study's requirements for elective courses about the Java, C++ and C# languages.

Covered by lectures

- NPRG005 Non-procedural Programming
- NPRG031 Programming II
- NSWI120 Principles of Computers
- NSWI154 Software Development Tools
- According to the choice of programming language:
    - C#
        - NPRG035 C# Language and .NET Framework
        - NPRG038 Advanced .NET Programming I
    - C++
        - NPRG041 Programming in C++
        - NPRG051 Advanced C++ Programming
    - Java
        - NPRG013 Java
        - NPRG021 Advanced programming for Java platform

5. Computer Architecture and Operating Systems

Computer architecture. Processors, multiprocessors. Buses, protocols. Input and output devices. OS architecture. Relationship between OS and HW architecture, interrupt handling. Processes, threads, scheduling. Synchronization primitives, mutual exclusion. Deadlock, deadlock recovery. Memory management, allocation algorithms. Principles of memory virtualization, paging, page replacement algorithms, page faults, page tables. File systems, directory structures.

- Computer architecture
    - representing data (bits, bytes, endianness, representing integers and real numbers, representing text and images, memory representation of structures)
    - bit operations and their usage (AND, OR, NOT, XOR, bit shifts)
    - von Neumann and Harvard architectures
    - single-chip computers
    - relationship of CPU, physical memory, devices
    - RAM vs. ROM/NVRAM (EPROM, EEPROM, flash)
    - address spaces
- Processors, multiprocessors

  - accumulator architecture, common register architecture, stack architecture
  - machine language (arithmetic and bit operations, memory operations, conditional and unconditional jumps, calling and returning from procedures and functions)
  - relationship of programming languages and machine language (compilation of common higher-level programming language constructs)
  - basic understanding of SMP multiprocessors (shared physical memory)
- Buses, protocols
  - serial vs. parallel buses
  - typical memory bus and typical system bus
    * knowledge of a specific bus is not required, only an understanding of necessary signals and their functions
    * address spaces and memory alignment
    * addressing devices
  - master vs. slave
  - basic understanding of USB buses
- Input and output devices
  - principle of communication with devices
  - memory-mapped I/O
  - HCI of a typical device (R/W, R/O, W/O registry, IRQ, DMA)
  - polling devices vs. IRQ
  - basic attributes of hard disks and SSD drives (division into sectors, speed and appropriate forms of access)

- OS architecture
  - booting a computer and operating system (computer firmware and its API, boot managers, boot loaders)
  - OS kernel, device drivers
  - supervisor and user mode of CPU and applications in a typical OS
  - binding of programming language to OS (OS API, programming language runtime, standard library of a programming language)
  - OS shell (function of shell, typical commands of Unix shell, shell scripts)
  - user management and user authentication in an OS
- Relationship between OS and HW architecture, interrupt handling
  - HAL
  - device drivers and driver stack
  - interrupt handling at the CPU and OS level
  - hardware exceptions and their handling and binding to programming language runtime (division by zero, etc.)
  - file descriptors and open files
  - file API, also as an abstraction for OS devices
- Processes, threads, scheduling
  - process concept in OS such as Unix or Windows (contents of process context, process hierarchy)
  - concept of threads (contents of thread context)
  - cooperative multitasking of threads
  - timers, interrupting timers
  - preemptive multitasking of threads
  - scheduler - basic concept, typical thread states (RUNNING, READY-TO-RUN, WAITING/SLEEPING, TERMINATED)
  - active vs. passive waiting (join, sleep, waiting for a lock, waiting for I/O operation to complete)
  - standard input and output and its redirection
  - pipes for interprocess communication
  - concept of UNIX signals
  - environment variables
- Synchronization primitives, mutual exclusion
  - race conditions and preventing them
  - critical sections and mutual exclusion
  - locks
  - deadlock and livelock (knowledge of the concept)

- Memory management, allocation algorithms
  - pointers and working with them, pointer arithmetic
  - stacks (usage and principle of allocation)
  - heaps (usage and basic idea of allocation principle)
- Principles of memory virtualization, paging, page replacement algorithms, page faults, page tables
  - virtual memory as a mechanism for protecting kernel and process memory
  - virtual memory vs. physical memory (pages vs. frames), swap files
  - single-level page tables (typical record structure)
  - page faults at the CPU level and their handling at the OS level

---

- File systems, directory structures
  - basic organization of file systems (boot sectors, metadata, data, free space - it is not necessary to know the details of any concrete FS)
  - text vs. binary files
  - directories and their hierarchy
  - file paths
  - symbolic links
  - typical metadata of files and directories
  - typical OS API for working with files and directories
  - current directory

Cancelled topics

- Recovery from deadlock

Covered by lectures

- NPRG030 Programming I
- NPRG031 Programming II
- NSWI095 Introduction to UNIX
- NSWI120 Principles of Computers
- According to the choice of programming language
  - C#
    - NPRG035 C# Language and .NET Framework
  - C++
    - NPRG041 Programming in C++
  - Java
    - NPRG013 Java

8. OI AO Algorithms and optimization

Approximation algorithms for combinatorial problems (satisfiability, independent set, set cover, scheduling). Applications of linear programming in approximation algorithms. The use of probability in the design of algorithms. Voronoi diagrams, arrangements of (complex) hyperplanes, incidence of points and lines, elementary computational geometry algorithms. Error-correcting codes. The probabilistic method - examples of application.

- Approximation algorithms for combinatorial problems
  - satisfiability
  - independent set
  - set cover
  - scheduling
- Applications of linear programming in approximation algorithms

- The use of probability in the design of algorithms

In addition to the knowledge of the individual algorithms, the topic also includes knowledge of general techniques used by the algorithms, and ability to apply these to similar problems (greedy algorithms, local search, applying linear programming, applying probabilistic methods).

Voronoi diagrams, arrangements of (complex) hyperplanes, incidence of points and lines, elementary computational geometry algorithms. Error-correcting codes. The probabilistic method - examples of application.

Covered by lectures

- NDMI009 Combinatorial and Computational Geometry I
- NDMI011 Combinatorics and Graph Theory I
- NOPT048 Optimization methods

8. OI DMS Discrete models and structures

Sets and mappings. Subvalence and equivalence of sets. Well ordering. Axiom of choice (Zermel's theorem, Zorn's lemma). Graph coloring (Brooks' and Vizing's theorem). Tutte's theorem. Extremal combinatorics (Ramsey's theorem, Erdös-Ko-Rado theorem). Error-correcting codes. The probabilistic method - application examples.

- Sets and mappings
  - overview of terminology (classes and proper classes, Cartesian product, relations, mappings, power set …)
  - Russell's paradox
- Subvalence and equivalence of sets
  - definition
  - Cantor-Bernstein theorem (without proof)
  - finite and infinite sets, definition of natural numbers
  - countable sets (definition, set operations that preserve countability) (without proofs)
  - Cantor's theorem (with proof by diagonal method)
  - cardinality of the sets of rational and real numbers (with main ideas of the proof of their non-equivalence)
  - the continuum hypothesis (formulation, independence from other axioms of set theory) (without proof)
- Well ordering
  - definition
  - relationship to proofs by mathematical induction
  - ordinal and cardinal numbers (definitions)
- Axiom of choice
  - equivalent formulations (non-emptiness of Cartesian product of non-empty sets, the well-ordering theorem (Zermelo's theorem), the maximum principle (Zorn's lemma), comparability of sets by cardinality) (without proof of their equivalence)
  - basic knowledge of consequences (every vector space has a basis, existence of non-measurable sets, Banach-Tarski paradox) (without proofs)
  - independence from other axioms of set theory (without proof)
- Graph coloring
  - definition and basic characteristics according to 14. Graph Theory
  - edge coloring (definition, formulation of Vizing's theorem, relationship to matching in a graph)
  - Brooks' theorem (formulation)
  - basic methods from the proofs of Vizing's and Brooks' theorems (Kempe chains, greedy algorithm)
- Matching in graphs
  - definition and basic characteristics according to 13. Discrete Mathematics
  - matching in general graphs (formulation of Tutte theorem including proof of simpler implication, Petersen's theorem and its proof using Tutte theorem)
  - Edmonds' algorithm (only knowledge of its existence)
- Extremal combinatorics

    - basic knowledge of what extremal combinatorics studies
    - Turán's theorem (formulation, Turán graphs)
    - Ramsey's theorem (formulation, proof for 2 colors)
    - Ramsey numbers (definition, upper bound for 2 colors from the proof of Ramsey's theorem and lower bound by probabilistic construction)
    - Erdös-Ko-Rado theorem (formulation)
- Error-correcting codes
    - overview of terminology
    - distance of codes and its relationship to the number of correctable and detectable errors
    - Hamming bound (formulation and proof)
    - perfect codes (definition and examples, Hamming code without precise construction)

Covered by lectures

- NAIL063 Set theory
- NDMI012 Combinatorics and Graph Theory II

8. SDI DW Databases and Web

Many-valued logic programming. Fagin's data model and algorithm. Success rates of algorithms. Framework for the transferability of information models. Information models and orderings.

Creating Web pages using HTML, XHTML and CSS. Server-side programming: PHP, Java, .NET. Client-side programming: JavaScript, AJAX. Web service technologies: REST, SOAP, WSDL. Linked Data: principles, RDF, RDF Schema. Principles of the XML format, defining the structure of XML data using DTD and XML Schema. DOM interface and SAX, XML Infoset. Query languages XPath, XQuery, XML Query Update. Transforming XML data using XSL. Databases with XML extensions and their principles, SQL/XML. Native XML databases and their principles. Overview of standard XML formats (DocBook, OpenOffice, SVG, XHTML, RDF, MathML, RSS, …)

Query optimization (indexes, hints, executions plans, data access methods, joining tables). Implementation of integrity constraints. Using XML extensions of relational databases. Full-text searching in SQL databases. File organization schemas. Basic types of indexes and their usage. B-trees and their variants. Hashing and secondary storage. Searching for data by multiple attributes. Mapping data structures into secondary storage. Text searching - Boolean and vector models. Hypertext searches, ranking, web page optimization for search engines. Similarity searching in multimedia databases. Metric indexing for similarity search. Collaborative filtering.

- Creating static web pages
    - Principles and syntax of markup languages (HTML, XHTML, HTML5) and their interpretation by a browser (visualization, DOM)
    - Influencing the appearance of Web pages using CSS (syntax, principle of operation, which appearance characteristics can be changed)
    - Basics of interaction with the user (links, forms)
- Architecture and basic principles of Web applications
    - CGI and CGI-like applications (principle of operation)
    - AJAX, single-page applications, relation to REST API
    - Connection to HTTP protocol, maintaining user sessions, cookies
- Basics of the PHP language and its usage in web applications
    - Basics of syntax and principles of a dynamic weakly-typed language
    - PHP HTTP wrapper and its functionality (automatic request processing, API for file upload)
- Client-side programming (JavaScript)
    - Basics of JavaScript syntax (ECMAScript), understanding of principles of prototype-based OOP nd using functional constructs (scope chaining, closure)
    - Working with documents through the DOM (overview of principles, basic knowledge of API)

    - Event handling in the DOM, event driven model, asynchronous programming in JavaScript
- Basics of web application security (authentication, authorization, encryption)
- Web service technology, principles of operation of REST, SOAP, WSDL
- Basics of semantic Web technology (Linked Data, RDF, RDF Schema)
- Semi-structured data formats
    - Syntax of XML and JSON
    - Defining schemas (DTD, XML Schema, JSON Schema), meaning of validation
    - Tools for working with XML (DOM, SAX)
    - Query languages XPath and XQuery (basic syntax)
    - XSL and transforming XML data using XSLT
    - XML databases (database with XML extensions, SQL/XML, native XML databases)

- Relational databases
    - Principles of relational databases, design of relational data models, primary keys, foreign keys
    - SQL language (basic syntax), detailed syntax of SELECT command (joining tables, aggregation, subqueries)
    - Meaning and usage of indexes, integrity constraints, SQL procedures and functions, triggers
    - SQL transactions and their usage, ACID, SQL isolation models
    - Full-text searching in SQL databases
- Implementing database systems
    - Mapping data structures to external storage
    - Implementation of basic types of indexes (B-trees and their variants, hashing in external storage)
    - Implementation of integrity constraints
    - Query optimization in database systems (hints, execution plans, data access methods, joining tables)
    - Implementing transaction processing (two-phase locking, timestamps, multiversioning)
- Searching in text
    - Boolean and vector models
    - Hypertext searches, ranking, optimizing Web pages for search engines
- Similarity searches in multimedia databases
    - Metric indexing for similarity search (clustering databases, filtering using pivots, metric trees)

Still to be expanded

- Many-valued logic programming
- Fagin's data model and algorithm
- Success rates of algorithms
- Framework for the transferability of information models
- Information models and orderings

Covered by lectures

- NDBI025 Database systems
- NDBI026 Database applications
- NDBI034 Multimedia Retrieval
- NSWI141 Introduction to Computer Networks
- NSWI142 Web applications
- NSWI166 Introduction to Recommender Systems

8. OI ML Computational linguistics

Formal languages and automata, basic formalisms for description of natural languages; morphological and syntactic analysis of a natural language; theory of information, language modeling.

- Formal languages and automata
    - Chomsky hierarchy of languages

    - regular languages and regular expressions
    - context-free languages
    - derivation trees
- Basic formalisms for description of natural languages
    - dependency and constituency trees
    - functional generative description
    - transformation grammar
    - unification grammars
- morphological and syntactic analysis of a natural language
    - morphemes, morphological analysis, part-of-speech tagging, stemming, lemmatization
    - checking for typing errors and grammar errors
    - syntactic analysis
    - lexical semantics, WordNet, supply of shared knowledge
- Language modeling
    - language corpora
    - basic principles of statistical modeling
    - Bayes' theorem
    - smoothing
- Basic information theory
    - entropy
    - conditional entropy and mutual information in discrete probability distributions

Covered by lectures

- NPFL012
- NPFL054
- NTIN071

6. OI Networking fundamentals

Taxonomy of computer networks. ISO/OSI reference architecture. Overview of the TCP/IP protocol model. Routing. Addresses, ports, sockets. Client-server architecture. Fundamentals of HTTP, FTP and SMTP protocols.

- Basic concepts and taxonomy of computer networks
    - Connection-oriented/connectionless, block-based/streaming, reliable/unreliable transmission of data
    - Circuit switching vs. packet switching, store-and-forward principle, best effort vs QoS
    - Categorizing networks by size, Internet, server networks, P2P networks
- ISO/OSI reference architecture and its binding to TCP/IP
    - Principle of layered architecture and its meaning
    - Functionality of individual layers, communication between layers
    - Meaning of ISO/OSI in modern networks built on TCP/IP
    - Examples of technologies/protocols commonly used in TCP/IP at individual layers
- Routing and IP protocol
    - Meaning of IPv4 protocol and IPv4 addresses
    - Algorithm for routing IP datagrams
    - Meaning of routing tables and ways to configure them
    - Typical problems in routing and delivering IP datagrams, ICMP protocol
- Transport layer
    - TCP and UDP protocols (basic overview and functionality)
    - Concept of a port and its meaning, concept of a communication connection, sockets
- Client-server architecture
    - Dependence on transport protocols
    - Meaning of this architecture from the standpoint of designing network applications and application protocols

- Most important applications of TCP/IP
  - HTTP
    - Principle of operation, use in applications
    - Message format, overview of important headers, methods of content encoding
    - Functionality of basic mechanisms (content negotiation, range serving, caching, sessions)
  - FTP
    - Principle of operation, active vs. passive mode, logging in vs. anonymous mode
    - Overview of basic sequences of commands and their operation
  - Electronic mail
    - Basic principles, terminology and protocols (mailbox, MTA, SMTP, POP3, IMAP)
    - Format of email messages, important headers
    - SMTP protocol, overview of basic commands, message forwarding
    - Spam and methods to defend against it (SMTP server configuration, sender identification, message filtering)

Future topics

- Basics of network communication security
  - Encryption (symmetric, asymmetric), how it is used in application protocols (HTTPS, SFTP) and electronic mail
  - Private and public keys, digital signatures, certificates

Covered by lectures

- NSWI045 TCP/IP Protocol Suite
- NSWI090 Computer Networks I
- NSWI141 Introduction to Networking

7. OI Optimization methods

> Polyhedron, Minkowski-Weyl theorem. Basics of linear programming, duality theorems, solution methods. Edmonds' algorithm. Integer programming.

TODO

Covered by lectures

- NOPT048 Optimization methods

8. PSS PG Computer graphics

> Raster images, colors, their perception, representation and reproduction, 2D and 3D drawing, basics of OpenGL, basics of realistic rendering and calculation of illumination, display functions, convolutions, Fourier transforms, radiometric correction, geometric correction, noise suppression in images, basics of pattern recognition.

- Raster and vector images
  - raster and vector principles
  - examples of concrete formats
  - principles of raster image compression
  - efficient image encoding (qtree)
  - partial transparency, alpha channel

- colors
  - basics of human visual system
  - color spaces (RGB, CMYK, HSV)
  - HDR graphics
  - basics of color reproduction
  - gamma correction
  - color palette reduction
- Rasterization
  - basics of raster line drawing
  - filling
  - anti-aliasing
- Basics of 3D graphics
  - matrix transformations
  - homogeneous coordinates
  - quaternions
  - animation curves
  - representing a 3D scene in a computer
  - scene hierarchy
- Real-time 3D graphics
  - GPU architecture
  - coordinate systems
  - basics of OpenGL (or Direct3D)
    - transferring data into GPU (buffers, textures, constants)
  - individual steps of a 3D pipeline
  - texturing
  - continuous shading techniques
  - shaders
  - typical architecture of a real-time 3D application (initialization, main loop)
- Photorealistic rendering
  - basic models of light reflection
  - principle of rendering equation
  - basics of ray-tracing
  - calculation of intersections and its acceleration
  - anti-aliasing and sampling
  - noise functions and their application
  - basics of Monte-Carlo rendering (distributed ray tracing, principle of path tracing)
- Image function - sampling
- Convolution
  - convolution theorem
  - inverse filter
  - Wiener filter
  - estimating parameters
- Fourier transform
  - definition in the 1D, 2D, continuous and discrete cases
  - characteristics of FT
  - FFT
  - uses for image processing
    - working with amplitude, phase, real and imaginary parts
    - filtering in the frequency domain
- Noise suppression in an image
  - linear and non-linear filters
  - edge detection
  - salt-and-pepper noise, white noise, noise parameters
  - elimination
    - convolution filters
    - frequency domain
    - averaging
- Radiometric correction
  - histogram
  - equalization by histogram
  - thresholding

  - reduction in brightness (color) level
- Geometric correction
  - basic geometric transformations in the plane
- Basics of pattern recognition
  - separable and non-separable classes
  - linear classifier
  - Bayes' rule

Covered by lectures

- NPGR002 Digital Image Processing
- NPGR003 Introduction to Computer Graphics
- NPGR004 Computer Graphics II

6. PSS+SDI Networking fundamentals

Taxonomy of computer networks. ISO/OSI reference architecture. Overview of the TCP/IP protocol model. Routing. Addresses, ports, sockets. Client-server architecture. Fundamentals of HTTP, FTP and SMTP protocols. Principles of data transmission - encoding, modulation, transmission speed/bandwidth, Nyquist-Shannon theorem, analog/digital transmission, transmission media. Data transmission technology - synchronous/asynchronous transmission, CRC, acknowledgements, flow control. Access methods, collision detection. Routing. Addresses, ports, sockets. QoS.

- Basic concepts and taxonomy of computer networks
  - Connection-oriented/connectionless, block-based/streaming, reliable/unreliable transmission of data
  - Circuit switching vs. packet switching, store-and-forward principle, best effort vs QoS
  - Categorizing networks by size, Internet, server networks, P2P networks
- Principles of data transmission
  - Basic concepts (encoding, modulation, transmission speed/bandwidth)
  - Nyquist-Shannon theorem and its meaning for data transmission
  - Analog/digital transmission, transmission media
  - Data transmission technology - synchronous/asynchronous transmission, CRC, acknowledgements, flow control
  - Access methods to shared media, collision detection
- ISO/OSI reference architecture and its binding to TCP/IP
  - Principle of layered architecture and its meaning
  - Functionality of individual layers, communication between layers
  - Meaning of ISO/OSI in modern networks built on TCP/IP
  - Examples of technologies/protocols commonly used in TCP/IP at individual layers
- Routing and IP protocol
  - Meaning of IPv4 protocol and IPv4 addresses
  - Algorithm for routing IP datagrams
  - Meaning of routing tables and ways to configure them
  - Typical problems in routing and delivering IP datagrams, ICMP protocol
  - Routing in the Internet, autonomous systems
  - IPv4 vs. IPv6 (significant differences)
- Transport layer
  - TCP and UDP protocols (basic overview and functionality)
  - Concept of a port and its meaning, concept of a communication connection, sockets
- Client-server architecture
  - Dependence on transport protocols
  - Meaning of this architecture from the standpoint of designing network applications and application protocols

---

- Most important applications of TCP/IP
  - HTTP

        - Principle of operation, use in applications
        - Message format, overview of important headers, methods of content encoding
        - Functionality of basic mechanisms (content negotiation, range serving, caching, sessions)
    - FTP
        - Principle of operation, active vs. passive mode, logging in vs. anonymous mode
        - Overview of basic sequences of commands and their operation
    - Electronic mail
        - Basic principles, terminology and protocols (mailbox, MTA, SMTP, POP3, IMAP)
        - Format of email messages, important headers
        - SMTP protocol, overview of basic commands, message forwarding
        - Spam and methods to defend against it (SMTP server configuration, sender identification, message filtering)

Future topics

- Basics of network communication security
    - Encryption (symmetric, asymmetric), how it is used in application protocols (HTTPS, SFTP) and electronic mail
    - Private and public keys, digital signatures, certificates

Covered by lectures

- NSWI045 TCP/IP Protocol Suite
- NSWI090 Computer Networks I
- NSWI141 Introduction to Networking

7. PSS+SDI Compilers

Structure of a compiler, lexical, syntactic analysis, generating intermediate code, optimization. Interpreted languages, virtual machines.

TODO

Covered by lectures

- NSWI098 Compiler Principles

8. PSS SIT Networks and Internet technologies

The TCP/IP protocol family (ARP, IPv4, IPv6, ICMP, UDP, TCP) - addressing, address assignment, translating between IP addresses and link-level addresses, routing, fragmentation, reliability, flow control, congestion control, NAT. The BSD sockets interface. Reliability - connection-oriented and connectionless protocols; types, detection and correction of errors. Security - IPSec, SSL, firewalls. Internet and intranet protocols and technologies - DNS, SMTP, IMAP, POP3, FTP, HTTP, NFS. IP telephony. 10-megabit, 100-megabit, gigabit versions of Ethernet, flow control. Wireless Ethernet (802.11 protocol architecture, access methods, roaming). xDSL. Mobile networks (GSM, 3G). Bluetooth.

This area implicitly assumes knowledge of 6. PSS+SDI Networking fundamentals.

- The most important TCP/IP protocols
    - Internet Protocol
        - IPv4 address meaning, private vs. public addresses, address assignment (allocation, DHCP)

        * Approximate format of an IPv4 datagram (important fields and their meaning), ensuring integrity, fragmentation, changes in an IPv6 datagram (as opposed to IPv4)
        * Dependence of IP on the network layer interface (meaning of IP over everything), converting network addresses (e.g. MAC) to IP adresses, ARP protocol
        * ICMP and its relation to IPv4, overview of most important ICMP messages
        * Address translation via NAT/NAPT, advantages and disadvantages of network translation, NAT tunneling (TURN, STUN)
        * Routing algorithm, configuring routing tables, internal vs. external routing, routing protocols (principle of operation of RIP, OSPF, BGP)
    - Transport protocol TCP/IP
        * Approximate format of TCP and UDP packets (important fields and their meaning)
        * Meaning of port, initiating a connection
        * Ensuring reliability in TCP, congestion control algorithm
        * BSD socket API and its relation to TCP and UDP
    - Domain Name System
        * Principle of operation, type of records, zones and delegation of authority
        * Algorithm for looking up DNS records over IP, caching in DNS, reverse records and reverse lookup

---

- Application network protocols
    - World Wide Web
        * Protocols HTTP, HTTP/2 and WebSocket (basic operation, meaning for web)
        * Dependence of WWW on DNS, meaning of URL and URI
        * Basics of HTML, meaning of a hyperlink, Linked Data
    - Electronic mail
        * SMTP, IMAP and POP3 protocols (basic operation, usage in electronic mail)
        * Message format, encoding of attachments (MIME), encrypted messages
    - File transfer protocols
        * FTP (principle of operation, overview of important commands, transfer modes, security)
        * NFS (basic principles of operation)
        * Samba (basic principles of operation)
    - Internet telephony
        * Principles of VoIP
        * Basics of SIP protocol
    - Remote access protocols (SSH, RDP)

- Security
    - IPSec (principles of operation, usage)
    - SSL/TLS and its usage in application protocols
    - Firewalls (principles of operation and configuration)
    - DNSSec

- Most common technologies at the network interface layer
    - Ethernet
        * Overview of basic characteristics (transfer medium, encoding, access method)
        * MAC adresses, frame format
        * Ethernet versions (10/100/1000Mb), most significant differences
    - Wireless Ethernet
        * 802.11 protocol architecture, communication band, meaning of WiFi designation
        * Access methods, roaming, security
    - xDSL
    - Mobile networks
    - Bluetooth

8. SDI SI Software Engineering

Many-valued logic programming. Fagin's data model and algorithm. Success rates of algorithms. Framework for the transferability of information models. Information models and orderings.

Creating Web pages using HTML, XHTML and CSS. Server-side programming: PHP, Java, .NET. Client-side programming: JavaScript, AJAX. Web service technologies: REST, SOAP, WSDL. Linked Data: principles, RDF, RDF Schema. Principles of the XML format, defining the structure of XML data using DTD and XML Schema. DOM interface and SAX, XML Infoset. Query languages XPath, XQuery, XML Query Update. Transforming XML data using XSL. Databases with XML extensions and their principles, SQL/XML. Native XML databases and their principles. Overview of standard XML formats (DocBook, OpenOffice, SVG, XHTML, RDF, MathML, RSS, …)

Design patterns. Important extensions and languages related to C++: C, C++/CLI, .NET, POSIX, GNU. Compile-time polymorphism: partial and explicit specializations, policy classes, traits. Concepts. Standard libraries: Containers, iterators, algorithms. Lambda expressions. Parallel environments. Safe and portable programming, preventing errors. Debugging errors and performance, advanced development tools. Program-environment interaction, OS interfaces. Typical paradigms of important interfaces: databases, XML, network communication, GUI, graphics.

- Creating static web pages
  - Principles and syntax of markup languages (HTML, XHTML, HTML 5) and their interpretation by a browser (visualization, DOM)
  - Influencing the appearance of Web pages using CSS (syntax, principle of operation, which appearance characteristics can be changed)
  - Basics of interaction with the user (links, forms)
- Architecture and basic principles of Web applications
  - CGI and CGI-like applications (principle of operation)
  - AJAX, single-page applications, relation to REST API
  - Connection to HTTP protocol, maintaining user sessions, cookies
- Basics of the PHP language and its usage in web applications
  - Basics of syntax and principles of a dynamic weakly-typed language
  - PHP HTTP wrapper and its functionality (automatic request processing, API for file upload)
- Client-side programming (JavaScript)
  - Basics of JavaScript syntax (ECMAScript), understanding of principles of prototype-based OOP nd using functional constructs (scope chaining, closure)
  - Working with documents through the DOM (overview of principles, basic knowledge of API)
  - Event handling in the DOM, event driven model, asynchronous programming in JavaScript
- Basics of web application security (authentication, authorization, encryption)
- Web service technology, principles of operation of REST, SOAP, WSDL
- Basics of semantic Web technology (Linked Data, RDF, RDF Schema)
- Semi-structured data formats
  - Syntax of XML and JSON
  - Defining schemas (DTD, XML Schema, JSON Schema), meaning of validation
  - Tools for working with XML (DOM, SAX)
  - Query languages XPath and XQuery (basic syntax)
  - XSL and transforming XML data using XSLT
- Design patterns
  - Overview of various design patterns, explanation of the principles of a given pattern and the situations, in which it is advantageous to use it
- Important extensions to C++
  - Compile-time polymorphism: partial and explicit specialization, policy classes, traits.
  - Standard libraries and their important elements (containers, iterators, algorithms)
  - Lambda expressions.
  - Parallel programming support, memory model
  - Threads, TLS, synchronization, atomic operations
- Safe and portable programming, preventing errors, debugging errors and performance, advanced develop-

ment tools
    - basic concepts of version control systems and scenarios for using them, basics of branching and merging for development on a team, cause and resolution of conflicts, centralized and distributed repositories
    - constructing large applications (dependency resolving mechanisms and portability)
    - basic concepts of functional testing, recommended methods for creating tests, looking for functional errors, recording events
    - performance measurement (basic concepts, recommended approaches)
    - generating documentation
    - basic approaches to debugging
    - integrated development environments
- Program-environment interaction, OS interfaces.
- Typical paradigms of important interfaces: databases, XML, network communication, GUI, graphics.

Still to be expanded

- Many-valued logic programming.
- Fagin's data model and algorithm.
- Success rates of algorithms.
- Framework for the transferability of information models.
- Information models and orderings.

Cancelled topics

- Important extensions and languages related to C++: C, C++/CLI, .NET, POSIX, GNU.

Covered by lectures

- NDBI034 Multimedia Retrieval
- NDBI037 Information Models with Ordering
- NPRG024 Design Patterns
- NPRG036 XML Technologies
- NPRG051 Advanced C++ Programming
- NSWI126 Advanced Tools for Software Development and Monitoring
- NSWI141 Introduction to Computer Networks
- NSWI142 Web Applications
- NSWI154 Software Development Tools

8. PSS SP System programming

Launching processes, dynamically linked libraries, calling conventions. Parallelism and synchronization on multiprocessors. Synchronization interfaces. Memory management on multiprocessors, allocators, garbage collection. Interfaces for working with files, memory-mapped files. Internal structure of fundamental system files. Design patterns. Version control. Testing functionality and performance. Middleware concept, calling remote procedures, message passing (concrete technologies according to current developments). Methods of specifying and verifying attributes of imperative programs.

- Launching processes, dynamically linked libraries, calling conventions
    - basic API for launching processes (fork, exec, join, CreateProcess, general understanding is OK)
    - relocation (absolute and relative addresses and the principle of address correction during relocation, PIC)
    - linking (purpose of linking, static and dynamic linking, concept of externs and exports)
    - calling conventions (basic elements of ABI, using registers, stack frame format, general understanding is OK)
- Parallelism and synchronization on multiprocessors

  - understanding and identifying individual race conditions in code
  - basic parallel concepts (processes, threads) on multiprocessors (general understanding is OK)
  - basic synchronization mechanisms on multiprocessors (IPI, T&S, CAS, LL/SC)
  - implementation of basic primitives (spin locks, semaphores) on multiprocessors
- Interfaces for synchronization
  - basic API and using atomic types, barriers, locks, semaphores, conditional variables
- Memory management on multiprocessors, allocators, garbage collection
  - API and functions of simple heap allocators (lists of blocks and their typical organization)
  - basic concepts of garbage collection (mutator, collector, reachability, garbage)
  - basic garbage collection algorithms (mark and sweep, copying collection)
  - generational hypothesis and usage for generational collection
- Interfaces for working with files, memory-mapped files
  - mechanisms of typical APIs for working with files and their motivation (opening files, scatter gather, async, map)
  - operation of memory-mapped files (concept, overall function of API, overall operation of VMM)
- Internal structure of basic system files
  - basic concepts used in managing disk space (sectors, blocks, trees, logs)
  - basic structures used for file recordkeeping (FAT, inode, extent)
  - using these structures in basic types of file systems (FAT, NTFS, ext, BTRFS, CD)

- Design patterns
  - purpose and usage of design patterns
  - knowledge and usage of basic design patterns
    - factory, builder, prototype, singleton
    - adapter, facade, proxy, decorator
    - observer, visitor, strategy, MVC

- Version control
  - concept of a version and history of versions, origin and resolution of conflicts
  - basic scenarios for version control systems
  - using branching and merging in development on a team
  - centralized and distributed repositories
  - basic knowledge of selected version control systems (Subversion, git)

- Testing functionality and performance
  - knowledge of basic concepts of functional testing
  - knowledge of basic concepts of performance profiling
  - basic knowledge of selected test frameworks (JUnit, MSTest)
  - recommended approaches for creating functional tests and performance profiling

---

- Middleware concept
  - typical architecture of distributed systems and the role of middleware
  - calling remote procedures
    - principle of RPC mechanism for functions and objects
    - serialization of arguments and of references
    - basic understanding of languages for describing function interfaces
    - basic understanding of selected technologies (protobuf, gRPC)
  - message passing
    - basic concepts of message passing (message, point to point, publish subscribe)
    - basic understanding of transfer protocols (unicast, multicast, especially in an IP network environment)
    - basic understanding of selected technologies (JGroups)

- Methods of specifying and verifying attributes of imperative programs
  - specifying program attributes using contracts (preconditions and postconditions)
  - partial and full program correctness

    – basic principles of verification in the PiVC system

Covered by lectures

- NSWI004 Operating systems
- NSWI154 Software Development Tools
- NSWI162 Program semantics
- NSWI163 Introduction to middleware