List of topics for the

FINAL EXAM

of the Software Information Technology BSc course of ELTE IK

Last modified: May 6, 2013

1. **Sequences, series, funcions, limits and continuity.** Convergence of sequences of real and complex numbers and vectors. Monotonic sequences. Notion and convergence of infinite numerical series. Positive series. The root and the ratio test. Alternating (Leibniz type) series. Power series. The Cauchy–Hadamard theorem. Limits and continuity of vector–vector functions. Properties of functions on a compact set: theorems of Heine, Weierstrass. Continuity of the inverse function. The Bolzano theorem.

2. **Differential and integral calculus and applications.** Differentiation and integration of functions of one real variable. Rules of differentiation. Theorems of Rolle, Cauchy and Lagrange. Extrema, concavity, discussion of functions. The Riemannian integral. Integration by parts, integration with substitution. The fundamental theorem of calculus (The Newton–Leibniz formula). Area, arc length, volume, surface. Differentiation of vector–vector functions. Jacobi matrix, gradient, partial derivative.

3. **Systems of differential equations.** The initial value problem. Linear systems of differential equations, linear equations of higher order.

4. **Methods of interpolation.** Lagrange interpolation, Hermite interpolation, spline interpolation.

5. **Numerical solution of linear systems of equations.** Gaussian elimination, methods based on product decomposition, iterative methods, least squares methods.

6. **Methods of number theory and criptography.** Number theoretical basics. RSA and applications, Diffie–Hellman–Merkle key exchange.

7. **Coding theory.** Huffmann code, error correcting codes, construction of finite fields. Reed–Solomon code and its deciphering.

8. **Basics of linear algebra.** Linear spaces, matrices, linear systems of equations. Linear programming, simplex algorithm.

9. **Basics of probability and statistics.** Discrete and continuous random variables, law of large numbers, central limit theorem. Statistical estimates, classical statistical tests.

10. **Artificial intelligence.** Pathfinding algorithms with relation to AI. State space representation. Description of search algorithms—local search, backtracking, graph search algorithms. Two-person games. Evolutionary algorithms, artificial neural networks. Automated logical deduction.

11. **Programming theorems.** The notions of programming theorems and reduction. Summation, counting, maximum search, conditional maximum search. Sequential search. Binary search. Problem solving using recursive functions. Program transformations and their importance in refining problems, and program efficiency.

12. **Type.** The notion of data type (type specification, representation, invariants, implementations). Type transformations of programming theorems to various objects of iterative structure (enumerable objects, e.g. vector, set, sequential input file). State space transform.

13. **Software development models.** Development stages of large scale systems. The concept of object oriented programming, type inheritance. Views of object oriented modeling, UML tools. The notion of software design patterns.

14. **Static and dynamic models.** Static model (class diagram, object diagram). Dynamic model (state diagram, sequence diagram, collaboration diagram, activity diagram). Use case diagrams.

15. **Compilation and execution of programs.** Compilers and interpreters, bytecode. Preprocessor, compiler, linker. Make. Compilation units, libraries. Syntactic and semantic rules. Static and dynamic type checking. Parallel programming.

16. **Data, operators and control structures.** Representation of numbers, basic data types. User defined types. Operators, expression evaluation. Statements, control structures, recursion, exception handling. Data abstraction. Class, inheritance, static and dynamic binding, subtype polymorphism. Generics.

17. **Program stucture.** Block, visibility, scope. Automatic, static and dynamic variables, garbage collection. Constructor, destructor. Cloning and comparing objects. Program modules, namespaces. Subroutines, parameter passing. Overloading.

18. **Compilers.** Structure of compilers, the tasks of the components. Lexical analyser and its functions, implementation. Classification and comparison of syntactic analysers, overview of their creation and functioning. The notion and use of ATGs. Code generation in assembly for basic imperative constructs.

19. **Logic.** The syntax and semantics of propositional and predicate calculus. The notion of logical consequence. Basic methods for proving logical consequences in propositional calculus: truth tables, the method of semantic tableaux, and the resolution.

20. **Theory of computation.** The Church–Turing thesis. Variants of Turing machines (one tape – multitape, deterministic – non-deterministic). Recursive and recursively enumerable languages. Undecidable problems. Reducibility. The time complexity classes P and NP. Polinomial time reducibility. NP-complete problems. The space complexity classes PSPACE and NPSPACE. Savitch's theorem. PSPACE-complete problems.

21. **Data structures.** Representations and operations of basic data structures—array, stack, queue, priority queue, list, binary tree, graph.

22. **Basic algorithms.** Representations of search data structures (binary search tree, AVL tree, 2-3 tree, B tree, hashing with chaining and open addressing). Sorting algorithms and their efficiency (bubble, insert, maximum selection, tournament, heap, quick and merge sort algorithms). Sorting in linear time: bucket sorts, radix sorts.

23. **Formal languages.** Generative grammars, the Chomsky hierarchy. Basic properties and application of regular and context-free grammars and languages. Syntax analysis of context-free languages. Finite automata and pushdown automata.

24. **Operating systems—parallel processes.** The notion and implementation of process and thread. Interactive, batch and real time processes, scheduling algorithms. Types of paralellism, race conditions, critical sections. Shared memory and messaging. Semafors and monitors. Deadlocks: description, avoidance, prevention, recognition.

25. **Operating systems—storage handling.** Hierarchy of storages. Memory handling: fixed and dynamic partitions, virtual memory. Paging and segmentation. Page replacement algorithms, working set. Scheduling input and output, reducing serving time. Organizing disk storage, physical and logical formatting, partitions. Redundant arrays, array handling systems. File systems and their implementation. Block reservation strategies, registering free storage space, journaled file systems.

26. **Computer networks.** Physical layer, data link layer, MAC, network layer, transport layer—services, methods, protocols.

27. **Distributed systems.** Specification of processes, properties of abstract parallel programs, solving representative problems with distributed and parallel programs. Or: properties of middle layers, naming conventions, communication, syncronization, consistency.

28. **Databases—query languages.** Relational model, entity–relationship model, transformation from ER to relational model. Relational algebra, SQL, Datalog. Recursion in query languages. Procedural elements in query langauges (variables, control structures, subprograms, cursors, exceptions).

29. **Databases—query execution.** Index structures, sparse and dense index, B+ tree, bitmap index, dynamic hashing. One-pass and two-pass algorithms, sort based and hash based algorithms. Join methods, cost of operations, query execution plans.