Thematic areas for the Master state exam in the subject Computer Science

I. Mathematical basics of computer science

1. Finite automata, regular expressions, closure properties of the class of regular languages.
2. Context-free grammars and languages. Pushdown automata, and their relation to context-free grammars.
3. Mathematical models of algorithms - Turing machines and Random Access Machines (RAM). Complexity of algorithms, asymptotic estimations. Undecidable problems.
4. Complexity classes; PTIME, NPTIME, and NP-complete problems.
5. Language of first-order predicate logic. Quantifiers and equivalent transformations of formulas.
6. Relation, operations with relations, properties of relations. Binary relations. Equivalence and partial order.
7. Operation and algebraic structures. Algebras with one or two binary operations.
8. FCA - formal context, formal concept, concept lattice.
9. Association rules, finding frequent itemsets.
10. Metric and topological spaces – metrics and similarities.
11. Clustering.
12. Random Variable. Types of Random Variable. Distribution Functions of Random Variable.
13. Discrete and Continuous Probability Distributions - Binomial, Hypergeometric, Negative Binomial, Poisson, Exponential, Weibull and Normal Distribution.
14. Exploratory Data Analysis. Descriptive Statistics and Graphs for Description of Qualitative and Quantitative Variables.
15. Inferential Statistics. Confidence Intervals. Principle of Hypothesis Testing.

II. Software engineering

1. Software process. Definition of software process, software process models, software process maturity.
2. Requirements engineering discipline. UML diagrams used in RE phase.
3. Definition of a discipline “Design”. UML diagrams used in this discipline. Design pattern – classification, description and examples.
4. Object oriented paradigm. Concept class, object, interface. Basic features of object and relation with class. Basic relations among classes and interfaces. Class vs. instance features.
5. Mapping of UML diagrams to source code.
6. Memory management (in languages C/C++, Java, C#, Python), virtual machine. Support for parallel execution, threads.
7. Error handling in modern programing languages. Principles of data streams – for input/output operation. Differences between character and byte oriented data streams.
8. Unified modeling language (UML) – types of diagrams and its usage during software development cycle.

III. Database and Information Systems (Data Processing Theory, Database and Information Systems)

1. Database systems modeling, conceptual modeling, data analysis, functional analysis.
2. Relational data model; function dependencies, decomposition and normal forms.
3. Transactions, recovery, log file, ACID, COMMIT and ROLLBACK operations; anomalies of concurrency, techniques and implementations: locking; isolation levels of transactions in SQL.
4. Procedural extensions of SQL: PL/SQL, triggers, cursors, bind variables, bulk operations.
5. Basic physical implementation of database systems: tables and indices; query execution plan.
6. Object-relational data model, XML data model: principles, query languages.
7. Data layer of information systems; API, frameworks and implementations; transactions in programming languages, security, object-relational mapping.
8. Distributed database management systems, fragmentation, replication.

IV. Computer Systems and Networks

1. Architectures of universal processors (CPUs). CPU acceleration techniques.
2. Microcomputers, basic construction features. Common integrated peripherals and their characteristics. Use cases of microcomputers.
3. Operating system structure and its relationship to computer hardware.
4. TCP/IP protocol family.
5. Media access control methods, shared medium, multiplexing.
6. Routing in computer networks (routing protocols, common problems). IP addressing. Network Address Translation.
7. Security in TCP/IP-based computer networks: possible attacks, packet filters, stateful firewall. Encryption and authentication, virtual private networks.

Last actualization 28.3.2018