Bachelor Exam Requirements
Applied Informatics

Software Engineering (SOFT)

Academic year 2025/2026

1. **Functional programming** (Programming paradigms – definition, main paradigms, fundamental differences. Basics of functional programming, referential transparency, reduction strategies, monomorphic and polymorphic data types, partial application of functions, intentional notation of lists and their utilization).

2. **Logic programming** (Basic concepts, backtracking, reversibility of logic programs, control of program flow, Prolog database. The main differences compared to other paradigms, the possibilities of using a logic programming language).

3. **Modeling of business processes** (meaning, basic concepts) BPMN (basic elements group, workflow patterns, diagrams types) Requirements, Use Cases (actors, Use case, scenarios, connections and relationships).

4. **Principle of Object Design** (development, procedural x object approach, object x class, abstraction, synergy, identity, responsibility, coherence, encapsulation, inheritance, polymorphism, domains, load).

5. **Classical and agile SW development methodologies** (general principles of classical methodologies, Waterfall model, Iterative development, RUP, Principles of Agile development, XP, Sprint, Scrum, Kanban, TDD).

6. **Formal representation of knowledge** (Data, information, knowledge, and their representation. Formal representation of knowledge, schemas, and languages, ontologies. Software support for designing, developing, and testing ontological models. Principles, tools, repositories, and uses of linked data).

7. **Semantic web** (Architecture, inference, and querying in the semantic web, query languages and tools. Systematic approach to semantic modeling, design patterns, development of semantic applications, methodologies. Practical applications of formal representation of semantics).

8. **Database architecture** (Database systems (DBS), Main functionalities of DBS. History of evolution of DBS. Data models. Relational algebra: projection, selection, and join functions. SQL).

9. **Conceptual modelling** (The E-R model and its graphical representation. Relational model. Types of relationships among entities and their representation in the relational model. Characteristics of a relational table. Normal forms of relational scheme).

10. **Mathematical principles of computer graphics** (Transformation, homogeneous coordinates, representation of transformations using matrices, their use and implementation, clipping, view frustum, basic transformation of the rendering pipeline, shape and position transformation, camera control, perspective solution).

11. **Rasterization** (Concepts of raster, pixel, data representation of a raster image, colour models and formats, rasterization algorithms of basic graphic entities and their implementation, image and object-oriented filling algorithms, pattern filling).

12. **Spatial scene rendering** (Visualization methods, algorithms for solving visibility and their complexity, calculation of surface illumination, calculation of shadows and light attenuation, scene representation,

vertex and index buffer, rendering pipeline, optimization procedures of 3D scene visualization, modern procedures for rendering, programable rendering pipeline and shaders, graphic libraries).

13. Image data (Raster image representation, sampling, quantization, aliasing and anti-aliasing methods, pre-processing methods, image filtering and types of raster filters, luminance function adjustment, geometric image transformation, interpolation, use of raster image histogram, image segmentation and classification, raster image compression methods, commonly used formats).

14. **Web applications** (Architecture of web applications, communication between client and server, requests and routing. Asynchronous transaction processing model. State management in web applications, methods, authentication and authorization. Risks and security of web applications, testing, logging).

15. **Inversion of Control (IoC)** (Principles, objectives and usage of such approach, relationship between IoC and Dependency Injection. Examples of usage, tools that support IoC).

16. **RESTful web services and microservices** (basic principles of web services and microservices, REST, security, SOAP).

17. **Architectural and design patterns** (service-oriented architecture (SOA), Model-View-Controller, Model-View-Presenter, Model-View-Template, Model-View-ViewModel, their interrelationships, Observer design pattern).

18. **Sorting algorithms** (characteristics and properties of sorting algorithms, selected sorting algorithms – Bubble sort, Insertion sort, Selection sort, Merge sort, Quick sort).

19. **Searching algorithms, text searching** (naive search, binary search, hashing. Issues of text search, Knuth Moris Pratt, Boyer-Moore, Karp-Rabin algorithms).

20. **Types of graphs, ways of graph representation, conceptual apparatus of graph theory, trees**, binary search tree, heap, spanning tree of the graph, number of spanning trees in selected graphs, minimum spanning tree (MST) in a graph, Jarník-Prim's, Kruskal's and Borůvka's algorithm for finding minimum spanning tree in a graph.

21. **Distances between vertices in a graph, Dijkstra's algorithm for finding the shortest paths between** nodes in a weighted graph , Labyrinth search, Tremaux's and Edmonds-Johnson's algorithms for labyrinth search, Eulerian graphs, Hamiltonian graphs, algorithms for finding Euler's path, breadth-first and depth-first graph search.

22. **Basic notions of AI** (AI definition, definition of algorithm, definition of heuristics, Turing test, Chinese room argument, semantic network, state representation of environment, breadth-first search, depth-first search, best-first search, A* algorithm, strong and weak AI, expert systems principle).

23. **Selected topics of AI** (functioning of artificial neuron, multiple-layer artificial neural networks with forward spread signal, backpropagation algorithm, application areas of artificial neural networks, genetic algorithm principle, operators of inheritance, crossover, mutation, fitness function, cognitive sciences, mental representations, mental procedures, CRUM conception).

Literature

Arlow, J., & Neustadt, I. (2005). UML 2 and the unified process: practical object-oriented analysis and design. Pearson Education.

Stellman, A., & Greene, J. (2014). Learning agile: Understanding scrum, XP, lean, and kanban. " O'Reilly Media, Inc.".

Silver, B., & Richard, B. (2009). BPMN method and style (Vol. 2). Aptos: Cody-Cassidy Press.

Levesque, H. J.: Thinking as Computation. MIT Press, 2017.

Lipovača, M.: Learn You a Haskell for Great Good!, No Starch Press, 2011.

Russell, S., Norvig, P. (2003): Artificial Intelligence – A Modern Approach (4th Global Edition). Pearson Ed., ISBN 978-0134610993.

SIKOS, Leslie. Mastering structured data on the Semantic Web: From HTML5 microdata to linked open data. Apress, 2015.

Sommerville, I. (2010). Software Engineering, 9"‘ed. England: Education Limited