Questions for the "Software and programming language theory" course (spring 2016)

Lecturers: V. A. Vasenin, M. A. Krivchikov

1. History of programming languages. Programming languages generations.
2. Programming languages classification. Type systems.
3. Programming languages classificaiton. Evaluation strategy.
4. Programming languages classification. Programming paradigms (functional, object-oriented, logical).
5. Syntax of programming languages. General notion. Lexical analysis. Chomsky hierarchy. BNF.
6. Syntax of programming languages. General notion. Alternative methods of specification: PEG, TDOP.
7. Syntax of programming languages. General notion. Implementation: attribute grammars, parser combinators.
8. Formal syntax of programming languages. Lexical and syntax analysis models.
9. Formal syntax. Total parser combinators.
10. Static semantics. Name binding and scoping.
11. Formal static semantics: mathematical models, implementation in proof assistants.
12. Static and dynamic semantics. Type systems of the programming languages. Type checking and type inference.
13. Type systems of the programming languages. Higher-order abstract syntax. Parametric higher-order abstract syntax.
14. Operational semantics. Structural (small-step) operational semantics.
15. Operational semantics. Natural (big-step) operational semantics.
16. Operational semantics. Implementation in proof assistants.
17. Denotational semantics. General notion. Meaning functions.
18. Denotational semantics. Domain theory. Fixed points.
19. Denotational semantics. Implementation: monads.
20. Axiomatic semantics. Hoare logic. Soundness and completeness.
21. Axiomatic semantics. Weakest common preconditions.