Core Courses
12x Courses
13x Courses
14x Courses

Diagram with nodes and OR connections.
15x Courses
16x Courses
17x Courses
18x, 19x Courses
Catalogue description. Introduces a broad range of theoretical topics in Computer Science, including the finite-state machine and regular languages, the pushdown automaton and context-free languages, the Turing machine and phrase structure languages, and the NP-complete complexity class. Fundamental limits on computation are developed.

Syllabus outline.

1. Deterministic and nondeterministic finite automata (FA)
2. Regular expressions
3. Pumping lemma for regular languages
4. Equivalence proof
5. Push-down automata (PDA)
6. Context-free languages (CFGs)
7. CKY parsing algorithm for CFGs
8. Deterministic and nondeterministic Turing machines
9. Decidability
10. Reducibility
11. The classes $\text{P}$ and $\text{NP}$
12. $\text{NP}$-completeness theorem
13. Additional $\text{NP}$-complete decision problems
14. The culture of theoretical computer science

Prerequisites. CS22

Textbooks Introduction to the Theory of Computation, Michael Sipser, Thomson Course Technology, 2006
CS 16: Introduction to Algorithms and Data Structures

Catalogue description. Introduces fundamental techniques for problem solving by computer that are relevant to most areas of computer science, both theoretical and applied. Algorithms and data structures for sorting, searching, graph problems, and geometric problems are covered. Programming assignments conform with the object-oriented methodology introduced in CS 15. Computer graphics animations are used to visualize algorithms and their performance.

Syllabus outline.

1. Analysis of Algorithms
2. Stacks
3. Queues, ArrayList
4. Sequences
5. Trees
6. Priority Queues
7. Heaps, Heapsort
8. Hashing
9. Dictionaries
10. 2-4 Trees
11. Red-black trees
12. Skip lists
13. Review
14. Quicksort
15. Mergesort
16. Radixsort
17. Text Searching
18. Convex hull
19. Tries, Data compression
20. Graphs
21. Depth-first search
22. Breadth-first search
23. Shortest path
24. Minimum spanning tree
25. Directed graphs
26. Maximum Flow
27. Numerical algorithms
28. Symmetric cryptosystems
29. RSA public-key cryptosystem
30. Digital signatures and cryptographic hash functions
31. Applications of cryptography

**Prerequisites.** CS 15 or written permission.

Catalogue description.

A continuation of CS17. Students learn to program in Java while continuing to develop their algorithmic and analytic skills. Object-oriented design of programs is a principal focus. Examples are drawn from such areas as discrete event simulation, window managers, computational biology and web clients/servers.

Syllabus outline.

1. The Java programming language
2. Objects, classes, interfaces, dynamic dispatch, subtyping, and inheritance
3. Iterators and streams
4. Multiple implementations of the list interface, stacks, queues, priority queues
5. Algorithms and data structures: in-place quicksort, list reversal, dynamic binary search tree (splay tree), BST node decorations, direct-address tables, hash tables, bucket sort, radix sort, graphs, Dijkstra’s algorithm, dynamic programming
6. Amortized analysis (e.g. growing and shrinking array)
7. Elementary probabilistic analysis (hashing)
8. Context-free grammars, simple parsing
9. Swing (Graphical User Interfaces)

Prerequisites. CS17

Textbooks