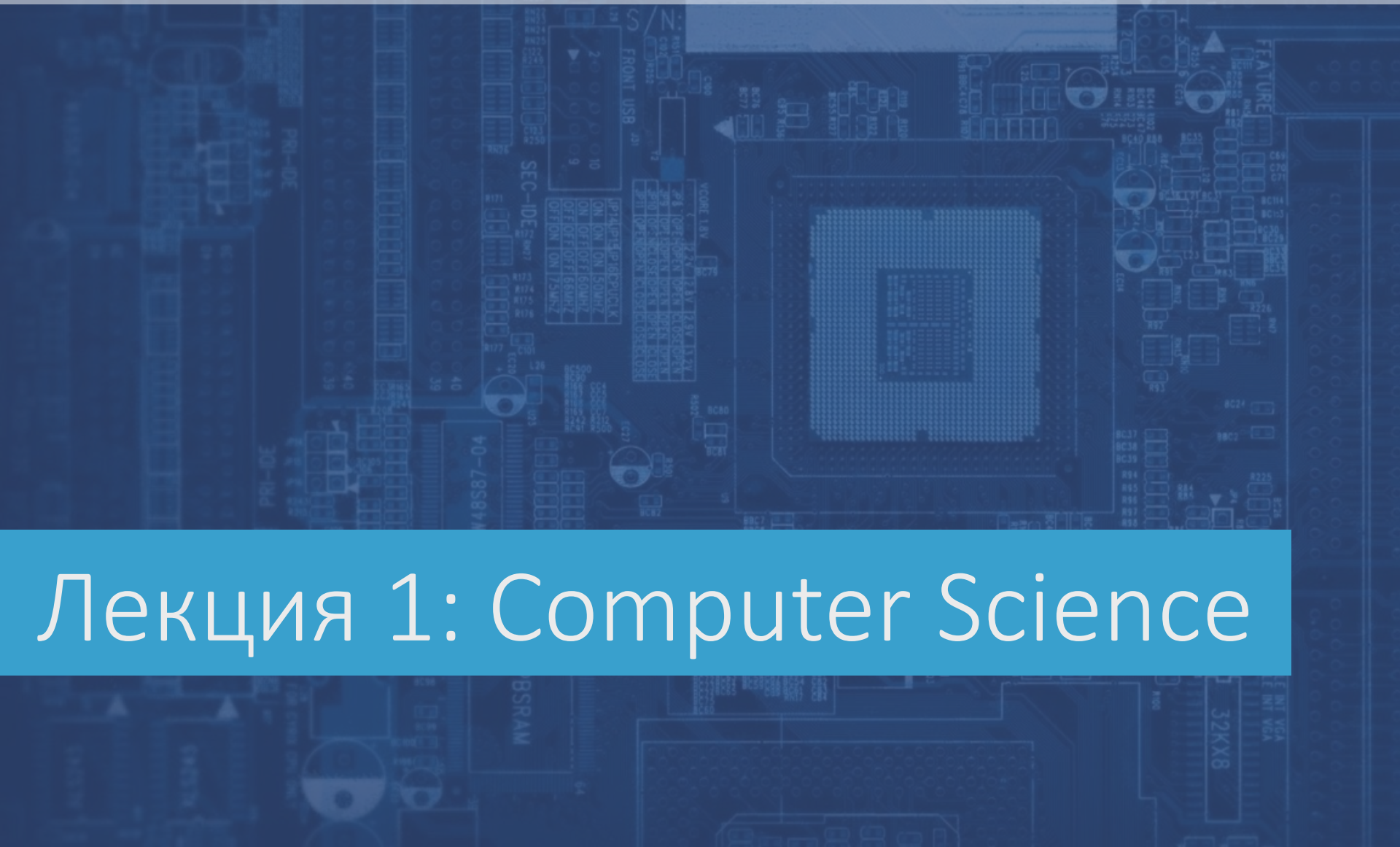
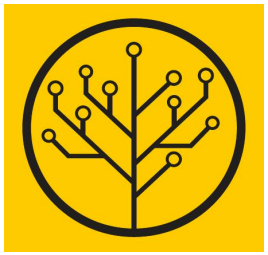
Научный семинар по
Компьютерным наукам



Лекция 1: Computer Science

Я хочу заниматься информатикой!

05.01.00 — инженерная и компьютерная графика.
05.01.00 — Инженерная геометрия и компьютерная графика.
05.02.00 — машиностроение и машиноведение, мехатроника, роботы, сварка[2]
05.03.00 — обработка материалов[3].
05.04.00 — энергетическое машиностроение, атомные реакторы, турбомашины[4].
05.05.00 — транспортное горное и строительное машиностроение[5].
05.07.00 — авиационная техника, летательные аппараты
05.08.00 — кораблестроение.
05.09.00 — электротехника, светотехника.
05.11.00 — приборостроение
05.12.00 — радиотехника, радиолокация, радионавигация.
05.13.00 — информатика, системный анализ.
05.14.00 — энергетика, электростанции.
05.15.00 — разработка и эксплуатация полезных ископаемых
05.16.00 — металлургия и материаловедение, нанотехнологии[6]
05.17.00 — химические технологии, полимеры и композиты
05.18.00 — пищевая промышленность, консервирование.
05.19.00 — лёгкая промышленность
05.20.00 — агроинженерные системы.
05.21.00 — деревообработка.
05.22.00 — транспорт, железные дороги, навигация, судовождение[7].
05.23.00 — строительство
05.24.00 — геодезия[8]
05.25.00 — документалистика
05.26.00 — безопасность труда
05.27.00 — электроника



ITMO UNIVERSITY

INNOPOLIS UNIVERSITY

Massachusetts
Institute of
Technology

- Программная инженерия
- Прикладная математика и информатика
- Бизнес-информатика
- Информатика и вычислительная техника
- Информационные системы и технологии
- Автоматизация технологических процессов и производств
- Компьютерная безопасность
- Информационная безопасность телекоммуникационных систем
- Приборостроение
- Мехатроника и робототехника
- Управление в технических системах
- Управление в технических системах
- Системы автоматического управления
- Приборы и системы ориентации, стабилизации и навигации
- Информационные системы и телекоммуникации
- Проектирование и технология производства электронной аппаратуры
- Системы обработки информации
- Компьютерные системы и сети
- Программное обеспечение ЭВМ и информационные технологии
- Информационная безопасность
- Высокопроизводительные компьютерные системы
- Защита информации
- Биотехнические системы и технологии
- Инфокоммуникационные технологии и системы связи
- Конструирование и технология электронных средств
- Материаловедение и технологии материалов
- Прикладная информатика
- Техносферная безопасность
- Электроника и наноэлектроника
- Управление качеством
- Прикладная математика


DETECTION
SOCIAL MEDIA
SERVICES
PROJECTS
WWW
BIG DATA
COMPUTING
INFORMATION
TECHNOLOGY
PROMOTION
MODELS
E-MARKETING
COMMUNICATION
MULTIMEDIA
NETWORK
CONTENT
PROCESSING
BRANDING
COMPUTER
CONSUMER
CONSUMER DEMAND
MARKETS
ORGANIZATION
PLANNING
SOFTWARE
CODING
WEB MARKETING
PREDICTIVE
PROGRAM
ANALYTICS
EVENTS
PROGRAMMING
DATA MINING
DATA SCIENCE
MACHINE LEARNING
VISION
ENGINEERING
RESEARCH
PROBABILITY
INFORMATION
WEB SERVICES
BIG
DATA
WEB DEV
KDD
COMPUTING
MATHS PATTERN
ENGINEERING
PROJECTS
SERVICE
STRATEGY
WORLDWIDE
VISUALIZATION
BIG DATA
SOLUTIONS
PLANNING
MEDIA
MOBILE
PRO
PRICING
SOCIAL NETWORK
SEGMENTATION
STATISTICS
TARGET
INFORMATION
DIGITAL
CODING
SOCIAL NETWORKS

Структура курса

- Обзорное изучение Python 3.5
- Лекции по различным разделам информатики
- Практика на Python

- Проектная работа (для желающих)
- Исследовательская работа (2 лекции за год)

- ДЗ – это **важно**!

Поехали!

Что такое computer science?

Что такое computer science? (это информатика :)

- **Computer science** (also called computing science) is the study of the theoretical foundations of information and computation and their implementation and application in computer systems.

- **Mathematical foundations**
  - **Coding theory** – Useful in networking and other areas where computers communicate with each other.
  - **Game theory** – Useful in artificial intelligence and cybernetics.
  - **Graph theory** – Foundations for data structures and searching algorithms.
  - **Mathematical logic** – Boolean logic and other ways of modeling logical queries; the uses and limitations of formal proof methods
  - **Number theory** – Theory of the integers. Used in cryptography as well as a test domain in artificial intelligence.

- **Algorithms and data structures**
  - **Algorithms** – Sequential and parallel computational procedures for solving a wide range of problems.
  - **Data structures** – The organization and manipulation of data.

- **Artificial intelligence**
  - **Artificial intelligence** – The implementation and study of systems that exhibit an autonomous intelligence or behavior of their own.
  - **Automated reasoning** – Solving engines, such as used in Prolog, which produce steps to a result given a query on a fact and rule database, and automated theorem provers that aim to provemathematical theorems with some assistance from a programmer.
  - **Computer vision** – Algorithms for identifying three-dimensional objects from a two-dimensional picture.
  - **Soft computing**, the use of inexact solutions for otherwise extremely difficult problems:
    - **Machine learning** - Automated creation of a set of rules and axioms based on input.
    - **Evolutionary computing** - Biologically inspired algorithms.
  - **Natural language processing** - Building systems and algorithms that analyze, understand, and generate natural (human) languages.
  - **Robotics** – Algorithms for controlling the behavior of robots.

- **Communication and security**
  - **Networking** – Algorithms and protocols for reliably communicating data across different shared or dedicated media, often including error correction.
  - **Computer security** – Practical aspects of securing computer systems and computer networks.
  - **Cryptography** – Applies results from complexity, probability, algebra and number theory to invent and break codes, and analyze the security of cryptographic protocols.

- **Computer architecture**
  - **Computer architecture –** The design, organization, optimization and verification of a computer system, mostly about CPUs and Memory subsystem (and the bus connecting them).
  - **Operating systems –** Systems for managing computer programs and providing the basis of a usable system.

- **Computer graphics**
  - **Computer graphics –** Algorithms both for generating visual images synthetically, and for integrating or altering visual and spatial information sampled from the real world.
  - **Image processing –** Determining information from an image through computation.

- **Concurrent, parallel, and distributed systems**
  - **Parallel computing -** The theory and practice of simultaneous computation; data safety in any multitasking or multithreaded environment.
  - **Concurrency (computer science) –** Computing using multiple concurrent threads of execution, devising algorithms for solving problems on multiple processors to achieve maximal speed-up compared to sequential execution.
  - **Distributed computing –** Computing using multiple computing devices over a network to accomplish a common objective or task and thereby reducing the latency involved in single processor contributions for any task.

- **Databases**
  - **Relational databases –** the set theoretic and algorithmic foundation of databases.
  - **Structured Storage -** non-relational databases such as NoSQL databases.
  - **Data mining –** Study of algorithms for searching and processing information in documents and databases; closely related to information retrieval.

- **Programming languages and compilers**
  - **Compiler theory –** Theory of compiler design, based on Automata theory.
  - **Programming language pragmatics –** Taxonomy of programming languages, their strength and weaknesses. Various programming paradigms, such as object-oriented programming.
  - **Programming language theory**
  - **Formal semantics –** rigorous mathematical study of the meaning of programs.
  - **Type theory –** Formal analysis of the types of data, and the use of these types to understand properties of programs — especially program safety.

- **Scientific computing**
  - **Computational science** – constructing mathematical models and quantitative analysis techniques and using computers to analyze and solve scientific problems.
  - **Numerical analysis** – Approximate numerical solution of mathematical problems such as root-finding, integration, the solution of ordinary differential equations; the approximation of special functions.
  - **Symbolic computation** – Manipulation and solution of expressions in symbolic form, also known as Computer algebra.
  - **Computational physics** – Numerical simulations of large non-analytic systems
  - **Computational chemistry** – Computational modelling of theoretical chemistry in order to determine chemical structures and properties
  - **Bioinformatics and Computational biology** – The use of computer science to maintain, analyse, store biological data and to assist in solving biological problems such as Protein folding, function prediction and Phylogeny.
  - **Computational neuroscience** – Computational modelling of neurophysiology.

- **Software engineering**
  - **Formal methods** – Mathematical approaches for describing and reasoning about software designs.
  - **Software engineering** – The principles and practice of designing, developing, and testing programs, as well as proper engineering practices.
  - **Algorithm design** – Using ideas from algorithm theory to creatively design solutions to real tasks.
  - **Computer programming** – The practice of using a programming language to implement algorithms.
  - **Human–computer interaction** – The study and design of computer interfaces that people use.
  - **Reverse engineering** – The application of the scientific method to the understanding of arbitrary existing software.

- **Web development**
  - **Web programming**
  - **Web designing**

- **Theory of computation**
  - **Automata theory** – Different logical structures for solving problems.
  - **Computability theory** – What is calculable with the current models of computers. Proofs developed by Alan Turing and others provide insight into the possibilities of what may be computed and what may not.
  - List of unsolved problems in computer science
  - **Computational complexity theory** – Fundamental bounds (especially time and storage space) on classes of computations.
  - **Quantum computing theory** – Explores computational models involving quantum superposition of bits.


python
TM

Материалы

- https://docs.python.org/3.5/tutorial/
- https://habrahabr.ru/post/150302/
- https://pythonworld.ru/samouchitel-python
- http://pythontutor.ru/

Среда разработки

- IDLE
- JetBrains PyCharm