

CONTENT

1. Administration of Operating Systems.....	2
2. Algorithms and data structures.....	4
3. Automata and formal languages.....	6
4. Classical and quantum computations.....	8
5. Computability theory.....	10
6. Computational complexity.....	12
7. Computer network Internet.....	14
8. Cryptographic systems and their applications.....	16
9. Data analysis project I.....	18
10. Introduction to cognitive and neural sciences.....	20
11. Introduction to machine learning.....	22
12. Introduction to neural networks.....	24
13. Linux basics.....	26
14. Logical aspects of databases.....	28
15. MATLAB and neurocognition.....	29
16. Parallel and distributed systems.....	31
17. Programming environments in schools II.....	33
18. Programming language Python.....	35
19. Project II.....	37
20. Software Systems Modelling.....	39
21. Technologies of big data processing.....	41
22. Typographical systems.....	43

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ AOS/25	Course name: Administration of Operating Systems
Course type, scope and the method: Course type: Practice Recommended course-load (hours): Per week: 2 Per study period: 28 Course method: present	
Number of ECTS credits: 3	
Recommended semester/trimester of the course:	
Course level: I., II., N	
Prerequisites:	
Conditions for course completion: To complete the course, students must successfully complete a project focused on configuring network services.	
Learning outcomes: The learning outcome is to understand the theoretical and practical background of Windows and Linux operating systems and selected network services.	
Brief outline of the course: 1) Linux Operating System Management (basic system tools for troubleshooting, system startup, network configuration), 2) File systems (general overview, RAID, LVM), 3) Container Management (Docker), 4) Web hosting services I (basic concept, APACHE), 5) Web hosting services II (SQL, HTTPS, security, NGINX), 6) File services (SAMBA, NFS, FTP), 7) Virtualization platforms (VMware, Proxmox), 8) Local network Management (routing, DHCP, firewall), 9) Remote device Management Automation (Ansible), 10) VPN, SSH, and Proxy, 11) Windows OS and Windows domain management, 12) Linux kernel, 13) Logging in Linux OS and Windows OS.	
Recommended literature: 1) LPIC-1 Exam 102. LPI [online]. Canada: The Linux Professional Institute, 2021 [cited 2021-9-22]. Available from: https://learning.lpi.org/en/learning-materials/102-500/ , 2) Linux - Documentation Project [online]. 4. Prague: Computer Press, 2007 [cited 2021-9-22]. Available from: https://i.info.cz/files/root/k/LDP_4.pdf , 3) The LPIC2 Exam Prep [online]. Sue B.V. - Open Sourced, 2021 [cited 2021-9-26]. Available from: https://lpic2book.github.io/src/	
Course language:	

Slovak or English					
Notes: Content prerequisites: Understanding basic concepts of operating systems, computer networks, and basic knowledge of the Linux shell (e.g., Bash) and PowerShell. The course is not organized every year.					
Course assessment Total number of assessed students: 55					
A	B	C	D	E	FX
70.91	14.55	7.27	0.0	5.45	1.82
Provides: doc. RNDr. JUDr. Pavol Sokol, PhD. et PhD., RNDr. Richard Staňa, PhD.					
Date of last modification: 05.11.2024					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ ASU1/15	Course name: Algorithms and data structures
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 2 / 1 Per study period: 28 / 14 Course method: present	
Number of ECTS credits: 4	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites: ÚINF/PAZ1a/15 and ÚINF/PAZ1b/15	
Conditions for course completion: Practice activities, homeworks and midterm exam. Final examination consisting of practice and theoretical test.	
Learning outcomes: Understand and learn algorithmic paradigms and data structures. Analyse time complexity of these algorithms.	
Brief outline of the course: Algorithms' time and space asymptotic complexity. Main Theorem. Amortized complexity. Brute Force. Backtrack. Divide and Conquer. Dynamic programming. Comparison and non-comparison sort algorithms. Sweep line algorithms. Graph Theory Algorithms. Data structures – queue, stack, priority queue, heap, prefix sum, binary search trees, interval trees, union & find, trie.	
Recommended literature: 1, Laaksonen A.: Guide to Competitive Programming: Learning and Improving Algorithms Through Contests (Undergraduate Topics in Computer Science), Springer, 2017, ISBN 978-3319725468 2, Forišek M., Steinová M.: Explaining Algorithms Using Metaphors. Springer Briefs in Computer Science, Springer (2013), ISBN 978-1-4471-5018-3 3, R. Sedgewick, K. Wayne: Algorithms (4th Edition), Addison-Wesley Professional, 2011, ISBN 978-0321573513, http://algs4.cs.princeton.edu/home/ 4, Open Data Structures: http://opendatastructures.org/	
Course language: Slovak or english	
Notes: Content prerequisites: - programming skills in some programming language (Python/Java/C++/...) - mathematics: -- computing with polynomials, logarithmic and exponential functions -- computing limits of sequences, L'Hospital rule	

Course assessment					
Total number of assessed students: 240					
A	B	C	D	E	FX
12.5	6.25	17.5	30.0	31.25	2.5
Provides: RNDr. Rastislav Krivoš-Belluš, PhD.					
Date of last modification: 08.01.2022					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ AFJ1a/15	Course name: Automata and formal languages
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 2 / 1 Per study period: 28 / 14 Course method: present	
Number of ECTS credits: 4	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites:	
Conditions for course completion: Oral examination.	
Learning outcomes: To provide theoretical background for studying computer science in general, by giving the necessary knowledge in theory of automata.	
Brief outline of the course: 1: Chomsky hierarchy of grammars: alphabet, symbol (letter, character), transitive closure, word (string), empty word (empty string), length of a string, concatenation, language, grammar, nonterminal symbol, terminal symbol, initial nonterminal (initial symbol), grammar rule, derivation step, language generated by a grammar, Chomsky hierarchy of grammars - phrase-structure, context sensitive, context free, regular 2: Deterministic finite state automata: finite state automaton, state, input symbol, output symbol, initial state, transition function, output function, examples of automata and their graphic representation, generalized transition and output functions and their basic properties 3: Reduction of automata I: equivalent automata, minimal (optimal) automaton, reachable state, properties of reachable states, elimination of unreachable states 4: Reduction of automata II: equivalent states, k-equivalent states, properties of equivalence and k-equivalence, relation between k-equivalence and (k+1)-equivalence, partitioning the state set into equivalence classes, elimination of equivalent states 5: Reduction of automata III: proof of correctness, unambiguity, and optimality of reduced automaton, testing equivalence of two automata 6: Deterministic finite state acceptors: basic definitions, language recognized by a finite state acceptor, common properties of acceptors and automata with an output, minimizing a finite state acceptor 7: Operations with regular languages: complement, intersection, union, difference, symmetric difference, testing of emptiness, inclusion, equality, and disjointness for regular languages 8: Nondeterministic finite state acceptors: definition, transition function, language recognized by a nondeterministic acceptor, elimination of nondeterminism 9: epsilon-acceptors: definition, properties, elimination of epsilon-transitions	

10: Regular grammars: regular grammar, extended regular grammar, transformation of acceptor to a regular grammar, transformation of extended regular grammar to an epsilon-acceptor
 11: Regular expressions I: basic properties, transformation of regular expression to an epsilon-acceptor
 12: Regular expressions II: regular equations, valid algebraic manipulations with regular expressions, solving an equation with a single unknown variable, solving a system of regular equations, transformation of acceptor to a regular expression
 13: Another constructions: review of transformations among various representations, an example of a direct transformation of a grammar to a regular expression, closure of the class of regular languages under another language operations – concatenation and Kleene star, mirror image
 14: Another operations: homomorphism and inverse homomorphism, a context-free language that is not regular

Recommended literature:

J.E. Hopcroft, R.Motwani, J.D. Ullman: Introduction to automata theory, languages, and computation, Addison-Wesley, 2001.
 J. Shallit: A second course in formal languages and automata theory, Cambridge University press, 2009.
 M. Sipser: Introduction to the theory of computation, Thomson Course Technology, 2006.

Course language:

Slovak or English

Notes:

Course assessment

Total number of assessed students: 971

A	B	C	D	E	FX
27.19	18.64	23.79	15.96	9.78	4.63

Provides: RNDr. Juraj Šebej, PhD., univerzitný docent

Date of last modification: 23.11.2021

Approved:

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ KKV1/21	Course name: Classical and quantum computations
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 3 / 2 Per study period: 42 / 28 Course method: present	
Number of ECTS credits: 6	
Recommended semester/trimester of the course:	
Course level: II., N	
Prerequisites:	
Conditions for course completion: Successful completion of the subject is conditioned by proper acquisition of basic concepts, algorithms and models and demonstrating the ability to apply them creatively. The acquisition of knowledge takes place: <ul style="list-style-type: none"> - continuously during the semester in the form of partial assignments, - a written test during the semester, - a written test at the exam, - oral exam. In order to receive an evaluation, it is necessary to obtain at least 50% of points from each of the three parts (assignments during the semester, written part of the exam, oral part of the exam). The detailed evaluation method is published in the AIS.	
Learning outcomes: By completing the subject, the student will get: <ul style="list-style-type: none"> - knowledge of the classification and design of probabilistic algorithms, - basic knowledge of the principles of quantum computers and their differences compared to classical computing models, - knowledge and skills about the design and functioning of quantum computing and become familiar with the most well-known algorithms, = basic quantum computer programming skills.	
Brief outline of the course: <ol style="list-style-type: none"> 1. Introduction to quantum quantum computers. Basics of classical complexity theory. 2. Boolean circuits and their basic properties. 3. Probability algorithms. 4. BPP class and probability testing. 5. Basic properties of circuits and Fermat's test. 6. Miller - Rabin's test and the position of the BPP class in the hierarchy of complexity models. 7. Introduction to quantum computing and mathematical foundations of quantum theory. 8. Spectral representation of self-adjoint operators. 9. Quantum states and Hilbert vector spaces. 10. Basic quantum operators and basic quantum algorithms. 	

11. Quantum teleportation, superdense coding and Grover's algorithm.
12. Fourier transformation.
13. Shor's algorithm.

Recommended literature:

1. BERMAN, G.P., DOOLEN, G.D., MAINIERI, R., TSIFRINOVIC, V.I. Introduction to Quantum Computers. World Scientific, 2003.
2. GRUSKA, J. Quantum Computing. McGraw-Hill, 1999.
3. JOHNSON, G. A Shortcut Through Time: The Path to the Quantum Computer, Knopf 2003.
4. KITAEV, A.Y., SHEN, A.H., VYALYI, M.N. Classical and Quantum Computation. American Mathematical Society, 2002.
5. NIELSEN, M.A., CHUANG, I.L. Quantum Computation and Quantum Information. Cambridge University Press, 2000.
6. HIRVENSALO, M., Quantum Computing, Springer 2004

Course language:

Slovak or english

Notes:

Content prerequisites:

Linear algebra, Group theory, Probability theory, Theory of algorithms, Introduction to quantum computers.

Course assessment

Total number of assessed students: 116

A	B	C	D	E	FX
27.59	37.93	15.52	6.03	3.45	9.48

Provides: prof. RNDr. Gabriel Semanišin, PhD., Mgr. Viktor Olejár

Date of last modification: 25.07.2022

Approved:

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ TVY/15	Course name: Computability theory
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 2 / 1 Per study period: 28 / 14 Course method: present	
Number of ECTS credits: 4	
Recommended semester/trimester of the course:	
Course level: I., II., N	
Prerequisites:	
Conditions for course completion: Two written examinations focused on the construction of Turing machines, creating sequences of (primitive) recursive functions, solving examples. Oral exam focused on the relationship between classes of recursive and computable functions, the problem of stopping a Turing machine.	
Learning outcomes: Knowledge of computational model of Turing machine, Goedelian arithmetization, and relationship between Turing computability and recursivity of functions.	
Brief outline of the course: <ol style="list-style-type: none"> 1. Turing machine, basic principles of work of Turing machine, formalization of basic notions 2. Shifting of states, compositions of machines, computations on composed machines 3. Modifications of configuration 4. Elementary Turing machines 5. Compositions of elementary Turing machines 6. Primitively recursive functions 7. Primitively recursive predicates 8. Functions and predicates from number theory 9. Goedelian arithmetizationa of Turing computability 10. Recursive functions 11. Relationship of recursivity and Turing computability 12. Halting problem 	
Recommended literature: <ol style="list-style-type: none"> 1. BRIDGES, Douglas. Computability, A Mathematical Sketch book. Springer--Verlag, 1994. ISBN:: 978-0387941745 2. BUKOVSKÝ, Lev. Teória algoritmov, ES UPJŠ, Košice, 1999. ISBN 8070973730 3. MACHTEY, Michael a Paul YOUNG. An Introduction to the General Theory of Algorithms, North--Holland, Amsterdam 1978. 4. KRAJČI, Stanislav. Teória vypočítateľnosti. http://ics.upjs.sk/~krajci/skola/vyucba/ucebneTexty/vypocitatelnost.pdf 	
Course language:	

Slovak					
Notes:					
Course assessment					
Total number of assessed students: 361					
A	B	C	D	E	FX
55.12	11.36	10.8	4.71	4.71	13.3
Provides: doc. RNDr. Ľubomír Antoni, PhD.					
Date of last modification: 04.01.2022					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ VYZ1/15	Course name: Computational complexity
Course type, scope and the method: Course type: Lecture Recommended course-load (hours): Per week: 2 Per study period: 28 Course method: present	
Number of ECTS credits: 4	
Recommended semester/trimester of the course:	
Course level: II., N	
Prerequisites:	
Conditions for course completion: Oral examination.	
Learning outcomes: To give students theoretical background in computational complexity and theory of NP-completeness.	
Brief outline of the course: 1: Introduction: the notion of computational complexity, computational time, computational model, example - the problem of sorting, computational complexity as an asymptotic function 2: Basic computational models: RAM and RASP computers, the cost of an elementary step on these computers, single-tape Turing machine, multi-tape Turing machine, nondeterministic variants of these computational models, transformations among these models with respect to the time complexity 3: The classes P and NP: basic definitions, presenting (un)undirected graphs on the input, 3COL – the set of all 3-colorable graphs is in NP, 2COL - the set of all 2-colorable graphs is in P, SAT – the set of satisfiable Boolean formulas is in NP, CNF-SAT - Boolean formulas in conjunctive normal form 4: Variants of P and NP: decision problem, the problem of finding a solution, optimization problem, polynomial conversions among different variants 5: NP-completeness: reducibility in polynomial time and its transitivity, definition of the NP-completeness and its basic properties 6: NP-completeness of SAT 7: Variants of SAT: 3CNF-SAT - satisfiability of Boolean formulas in 3-conjunctive normal form, kCNF-SAT, CNF-SAT - satisfiability in k-conjunctive (conjunctive) normal form, 2CNF-SAT is in P 8: 3COL and its variants: 3COL (the problem of coloring vertices of a graph with 3 colors) in NP-complete, consequently: for each $k > 3$, kCOL (the problem of coloring with k colors) is NP-complete as well 9: Colorability of a planar graph with three colors: presenting a planar graph on the input, the proof of NP-completeness, coloring with a larger number of colors 10: Another NP-complete problems: Exact set cover, Clique, Vertex cover	

11: Hamiltonian path: Hamiltonian path in a directed and in undirected graph
 12: Subset-sum-like problems: Subset Sum - the problem of whether any subset of the integers sum to precisely a target sum, Partition - the problem of whether a given multiset of positive integers can be partitioned into two subsets with equal sums, a “more relaxed” version of Partition - achieving an approximate equality of the sums, distribution of tasks among K parallel processors
 13: Beyond P a NP: a review of the basic complexity classes - L, NL, P, NP, PSpace , NPSpace, ExpTime, NExpTime, ..., simulation of (non)deterministic space in (non)deterministic time, conversions in opposite directions
 14: PSpace: QBF - true quantified Boolean formulas, prenex normal form, Pspace completeness of QBF, PSpace = NPSpace

Recommended literature:

1. J.E. Hopcroft, R.Motwani, J.D. Ullman: Introduction to automata theory, languages, and computation, Addison-Wesley, 2007.
2. M. Sipser: Introduction to the Theory of Computation, Thomson, 2nd edition, 2006.
3. L.A.Hemaspaandra, M.Ogihara: Complexity theory companion, EATCS series, texts in computer science, Springer-Verlag, 2002.
4. S. Arora, B. Barak: Computational Complexity: A Modern Approach, Cambridge Univ. Press, 2009.
5. G.Brassard, P.Bradley: Fundamentals of algorithmics, Prentice Hall, 1996.
6. D.P.Bovet, P.Crescenzi: Introduction to the theory of complexity, Prentice Hall, 1994.
7. C. Calude and J. Hromkovič: Complexity: A Language-Theoretic Point of View, in G. Rozenberg and A. Salomaa, Handbook of Formal Languages II, Springer, 1997.

Course language:

Slovak or english

Notes:

Content prerequisites:

Basic notions from the theory of automata and formal languages.

Basic skills in programming and design of algorithms (in any programming language).

Basics knowledge in mathematical logic, set theory, and graph theory.

Course assessment

Total number of assessed students: 423

A	B	C	D	E	FX
57.92	15.37	12.77	6.62	7.09	0.24

Provides: doc. RNDr. Ondrej Krídlo, PhD.

Date of last modification: 03.09.2025

Approved:

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ PSIN/15	Course name: Computer network Internet
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 3 / 1 Per study period: 42 / 14 Course method: present	
Number of ECTS credits: 5	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites: ÚINF/PAZ1a/15 or ÚINF/PRG1/15	
Conditions for course completion: Activity at excercises (max 18 points), home work (max 18 points), test (max 30 points). Verbal exam (min 25 points, max 50 points). Required minimum for passing the course is 55 points.	
Learning outcomes: Students will get the informations about principles and achitecture of Internet. They will understand the principles of ISO/OSI layers reference model for network communication. They will understand the meaning and usage of terms protocol, service, interface. They will analyze the parameters of communication channels, understand the function of interconnection devices (hub, switch, router). They will understand the structure of IP packets, addressing and how packets are transmitted, the principle of routing protocols and the creation of routing tables. They will understand the priciples of acknowledged TCP transport transmission and its implementation. They will know how to use the interface of UDP and TCP protocols in a program code. They will understand the basic application protocols of the Internet.	
Brief outline of the course: 1. Introduction to computer networks, internet connection types, delay and loss in packet-switched networks, ISO OSI reference model and TCP/IP protocols family. 2. Application layer: Web and HTTP, protocol FTP ,e-mail and protocols SMTP, POP3, IMAP, 3. Application layer: domain names and DNS, Peer-to-peer applications. Security in computer networks. 4. Transport layer: services, multiplexing and demultiplexing, protocol UDP, reliable data transfer 5. Transport layer: connection oriented transport protocol TCP, flow and congestion control. 6. Network Layer: Internet protocol IPv4, virtual circuit and datagram networks, packet fragmentation, routing table, application protocol DHCP 7. Network Layer: network address translation NAT, ICMP protocol, internet protocol IPv6 8. Network Layer: routing algorithms and protocols, broadcast and multicast routing 9. Link layer: error detection, multiple access methods CSMA/CD and CSMA/CA, Ethernet, frames, protocols ARP and RARP, link layer addressing 10. Link Layer and wireless and mobile networks: hub, switch, virtual LAN, 802.11 Wireless LAN, Bluetooth 802.15, WiMAX 802.16, Mobile IP, mobility in GSM 11. Physical Layer: Communication channels parameters, digital and analog encoding.	

Recommended literature:					
1. J. F. Kurose, Keith W. Ross: Computer Networking: A Top-Down Approach, 7. edition, 2016					
2. A. S. Tanenbaum: Computer Networks, 5. edition, Pearson, 2010					
3. W. Stallings: Local and Metropolitan Area Networks, Prentice Hall, 2000					
4. E. Comer, R.E. Droms: Computer Networks and Internets, Prentice Hall, 2003					
5. W. R. Stevens: TCP/IP Illustrated, Vol.1: The Protocols, Addison-Wesley, 1994					
Course language:					
Slovak or English					
Notes:					
Content prerequisites: basic programming skills in Java					
Course assessment					
Total number of assessed students: 374					
A	B	C	D	E	FX
10.7	9.09	18.98	20.05	29.14	12.03
Provides: RNDr. Peter Gurský, PhD., univerzitný docent, RNDr. Richard Staňa, PhD.					
Date of last modification: 04.01.2022					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ KRS/15	Course name: Cryptographic systems and their applications
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 3 / 2 Per study period: 42 / 28 Course method: present	
Number of ECTS credits: 6	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites:	
Conditions for course completion: Homeworks, midterm written exam, active participation in laboratory exercises. Final written exam, possibly oral exam.	
Learning outcomes: This course covers the basic knowledge in understanding and using cryptography. The main focus is on definitions, theoretical foundations, and rigorous proofs of security, with some programming practice. Topics include symmetric and public key encryption, message integrity, hash functions, block cipher design and analysis, number theory, and digital signatures. The course also provides an introduction to cryptographic protocols for authentication and key management, including PKI and certificates.	
Brief outline of the course: Classical cryptography, basic information theory, cryptanalysis, security of classical ciphers. Symmetric ciphers - stream ciphers, block ciphers (DES, AES), modes of operation. Asymmetric ciphers - RSA, Elgamal, elliptic curve cryptosystems. Hash functions, message authentication codes, digital signatures. Authentication, key establishment and distribution, certificates.	
Recommended literature: 1. PAAR, Ch., PELZL, J.: Understanding Cryptography, Springer 2010. 2. STINSON, D. R., PATERSON, M. B.: Cryptography: Theory and Practice. CRC Press, 2018. 3. MAO, W. Modern Cryptography: Theory and Practice. Prentice Hall, 2003. 4. MENEZES, A., OORSCHOT, P. van, VANSTONE, S.: Handbook of Applied Cryptography. CRC Press, 1996. 5. SCHNEIER, B.: Applied Cryptography, 20th Edition, John Wiley & Sons Inc., 2015	
Course language: Slovak or English	
Notes: Content prerequisites: basic number theory and algebra, basic programming	

Course assessment					
Total number of assessed students: 146					
A	B	C	D	E	FX
15.75	8.22	14.38	15.75	32.19	13.7
Provides: doc. RNDr. Jozef Jirásek, PhD., RNDr. Rastislav Krivoš-Belluš, PhD.					
Date of last modification: 08.01.2022					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ PDA/19	Course name: Data analysis project I
Course type, scope and the method: Course type: Practice Recommended course-load (hours): Per week: 4 Per study period: 56 Course method: present	
Number of ECTS credits: 4	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites:	
Conditions for course completion: Active presentation of the results of the data analysis project and the bachelor thesis. Presentation of results from published foreign papers.	
Learning outcomes: Knowledge and skills associated with the phases of data analysis and their use in solving specific tasks. Knowledge of selected procedures in the field of data analysis, machine learning and artificial intelligence. Development of understanding of professional text in the field of data analysis and its interpretation.	
Brief outline of the course: 1. - 2. Phases of data analysis projects and selected basic terms 3. - 4. Selection and specification of data analysis projects 5. - 7. Selected methods of data analysis 8. - 11. Consultations on data analysis projects and bachelor theses 12. - 13 Applications of data analysis methods in various fields	
Recommended literature: 1. AGGARWAL, Charu C. Data mining: a textbook. Cham: Springer, 2015. ISBN 978-3-319-14141-1. 2. ALPAYDIN, Ethem. Introduction to machine learning. 3rd ed. Massachusetts: MIT Press, 2014. ISBN 978-0-262-02818-9. 3. RASCHKA, Sebastian, Mirjalili, Vahid. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, 3rd Edition, Packt Publishing Ltd., 2019. ISBN 978-1789955750. 4. WITTEN, I. H., Eibe FRANK a Mark A. HALL. Data mining: practical machine learning tools and techniques. 4th ed. Amsterdam: Morgan Kaufmann, 2017. Morgan Kaufman series in data management systems. ISBN 9780128042915. 5. Literature associated with particular project.	
Course language: Slovak or English	
Notes:	

Course assessment					
Total number of assessed students: 37					
A	B	C	D	E	FX
94.59	0.0	0.0	0.0	5.41	0.0
Provides: doc. RNDr. Ľubomír Antoni, PhD.					
Date of last modification: 25.01.2022					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ UKN/24	Course name: Introduction to cognitive and neural sciences
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 2 / 2 Per study period: 28 / 28 Course method: present	
Number of ECTS credits: 5	
Recommended semester/trimester of the course:	
Course level: I., II., N	
Prerequisites:	
Conditions for course completion: Midterm exam Final exam consisting of written and/or oral part	
Learning outcomes: Overview anatomy, physiology, and cognitive processes in the human brain with focus on computational aspects of cognition and computational tools used in neuroscience.	
Brief outline of the course: 1. Intro to neural and cognitive science 2. Overview of anatomy and physiology of the central nervous system (CNS) 3. Methods of study in neuroscience. Sensory, motor and associative brain areas. 4. Neuron: anatomy, types, action potential 5. Propagation of signals in the neuron, neural coding. 6. Synaptic transmission and plasticity - neural basis of learning and memory. 7. Psychology of memory and learning. 8. Vision: Intro. Perception of brightness, edges, color. Model BCS/FCS. Perception of size and sitance. 9. Hearing and auditory cognition. 10. Language, psycholinguistics, speech perception and production. 11. Attention. 12. Crossmodal interaction (vision, hearing, touch). 13. Reasoning and decision making.	
Recommended literature: 1. Poeppel D., Mangun G., Gazzaniga M. (ed.): The Cognitive Neurosciences. 6th ed. MIT Press. 2020. ISBN-13: 978-0262043250 2. Dayan P and LF Abbott: Theoretical Neuroscience - Computational and Mathematical Modeling of Neural Systems. MIT Press, 2005 ISBN-13: 978-0262541855 3. Thagard P: Mind: Introduction to Cognitive Science, 2nd Edition. Bradford Books. ISBN-13: 978-0262701099	
Course language:	

Slovak or English					
Notes: Content prerequisites: Algebra, programming (Matlab).					
Course assessment Total number of assessed students: 14					
A	B	C	D	E	FX
42.86	0.0	14.29	7.14	35.71	0.0
Provides: doc. Ing. Norbert Kopčo, PhD., univerzitný profesor, Ing. Peter Lokša, PhD., RNDr. Keerthi Kumar Doreswamy, PhD., Ing. Udbhav Singhal, PhDr. Myroslav Fedorenko					
Date of last modification: 19.03.2024					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ USU/19	Course name: Introduction to machine learning
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 2 / 2 Per study period: 28 / 28 Course method: present	
Number of ECTS credits: 5	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites:	
Conditions for course completion: Creating a project focused on the application of machine learning algorithms in a selected application domain. Continuous written work focused on the preparation, processing and interpretation of data using machine learning methods. Successful completion of an oral exam focused on selected machine learning methods.	
Learning outcomes: Theoretical knowledge in the area of machine learning. Basic concepts of machine learning. Basic machine learning algorithms.	
Brief outline of the course: 1. Basic concepts of machine learning. 2. Basic characteristics of data, types of attributes, characteristics for individual attributes, dependence between attributes. 3. Data sources and their acquisition. Determining the target task. 4. Preparation and cleaning of data, missing values, incorrect inputs. 5. Classification tasks 6. Selected classification methods 7. Evaluation of models - true positive, false positive, true negative, false negative examples. 8. Classification accuracy indicators. 9. Cluster analysis. 10. Association rules. 11. Prediction tasks and selected prediction methods 12. Prediction accuracy indicators.	
Recommended literature: 1. AGGARWAL, Charu C. Data mining: a textbook. Cham: Springer, 2015. ISBN 978-3-319-14141-1. 2. ALPAYDIN, Ethem. Introduction to machine learning. 3rd ed. Massachusetts: MIT Press, 2014. ISBN 978-0-262-02818-9. 3. RASCHKA, Sebastian, Mirjalili, Vahid. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, 3rd Edition, Packt Publishing Ltd., 2019. ISBN 978-1789955750.	

4. WITTEN, I. H., Eibe FRANK a Mark A. HALL. Data mining: practical machine learning tools and techniques. 4th ed. Amsterdam: Morgan Kaufmann, 2017. Morgan Kaufman series in data management systems. ISBN 9780128042915.					
Course language: Slovak or English					
Notes: Content prerequisites: Basics of programming in Python, or another alternative programming language suitable for data analysis					
Course assessment Total number of assessed students: 73					
A	B	C	D	E	FX
84.93	6.85	4.11	2.74	1.37	0.0
Provides: doc. RNDr. Ľubomír Antoni, PhD., Mgr. Barbora Ľapinová					
Date of last modification: 20.09.2021					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ UNS1/15	Course name: Introduction to neural networks
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 2 / 2 Per study period: 28 / 28 Course method: present	
Number of ECTS credits: 5	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites: ÚINF/PAZ1a/15 or ÚINF/PRG1/15	
Conditions for course completion: The condition for passing the course is the realization of a project with the application of neural networks, successful completion of two written tests in the field of neural networks, their basic types, and genetic algorithms, as well as successful completion of the written and oral part of the exam.	
Learning outcomes: The result of the education is an understanding of the basic principles of neural networks and genetic algorithms. The student will gain the ability to apply the acquired knowledge in intelligent data analysis and also work with a selected tool for modeling neural networks.	
Brief outline of the course: 1. Basic concept arising from biology. Linear threshold units, polynomial threshold units, functions calculable by threshold units. 2. Perceptrons. Linear separable objects, adaptation process (learning), convergence of perceptron learning rule, higher order perceptrons. 3. Forward neural networks, hidden neurons, adaptation process (learning), backpropagation method. 4. Recurrent neural networks. Hopfield neural networks, properties, associative memory model, energy function, learning, optimization problems (business traveler problem). 5. Model of gradually created network. ART network, architecture, operations, initialization phase, recognition phase, search and adaptation phase. Use of the ART network. 6. Applications of studied models in solving practical problems. 7. Written test I. 8. Motivation to model genetic elements. Genetic algorithm. Application of genetic algorithms. 9. Genetic programming, root trees, Read's linear code. Basic stochastic optimization algorithms: blind algorithm and climbing algorithm. Forbidden search method. 10. Genetic and evolutionary programming with typing, examples of use. Grammatical evolution. 11. Special techniques of evolutionary computations. Selection mechanisms in evolutionary algorithms. 12. Use of genetic algorithms in training neural networks. Artificial life. 13. Written test II.	

Recommended literature:

1. AGGARWAL, Charu C. Neural networks and deep learning: a textbook. Cham: Springer, 2018. ISBN 978-3319944623.
2. KVASNIČKA, Vladimír. Úvod do teórie neurónových sietí. [Slovenská republika]: IRIS, 1997. ISBN 80-88778-30-1.
3. KVASNIČKA, Vladimír. Evolučné algoritmy. Bratislava: Vydavateľstvo STU, 2000. Edícia vysokoškolských učebníc. ISBN 80-227-1377-5.
4. MITCHEL, Melanie. An Introduction to Genetic Algorithms. Cambridge: MIT Press, 2002. ISBN 0-262-63185-7.
5. SINČÁK, Peter, ANDREJKOVÁ, G. Úvod do neurónových sietí, I. diel, Košice: ELFA, 1996. ISBN 808878638X

Course language:

Slovak or English

Notes:

Content prerequisites:

Basics of programming in Python, or another alternative programming language suitable for data analysis

Course assessment

Total number of assessed students: 567

A	B	C	D	E	FX
26.81	16.23	19.4	16.05	18.17	3.35

Provides: doc. RNDr. Ľubomír Antoni, PhD., RNDr. Šimon Horvát, PhD.**Date of last modification:** 23.11.2021**Approved:**

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice					
Faculty: Faculty of Science					
Course ID: ÚINF/ ZLI/26		Course name: Linux basics			
Course type, scope and the method: Course type: Practice Recommended course-load (hours): Per week: 2 Per study period: 28 Course method: present					
Number of ECTS credits: 2					
Recommended semester/trimester of the course:					
Course level: I., N					
Prerequisites:					
Conditions for course completion: The condition for passing the course is: 1. Homeworks (50% of the total number of points), 2. Written final theoretical exam (25% of the total number of points), 3. Written final practical exam (25% of the total number of points).					
Learning outcomes: The result of the education is an understanding of the theoretical and practical background for studying computer science, by giving the necessary knowledge in the usage of Unix/Linux operating systems.					
Brief outline of the course: 1. Introduction to Unix/Linux systems, 2. Linux ommand line, 3. Text processing tools, 4. Managing files, 5. Managing users, groups and rights, 6. Managing processes, 7. Managing software and packages, 8. Administering the system - system booting, jobs, logging,9. Basic networking, 10. Managing network interfaces, 11. Managing disk partitions, 12. Exam.					
Recommended literature: 1. LPIC-1 Exam 101. LPI [online]. Canada: The Linux Professional Institute, 2021 [cit. 2021-9-22]. Dostupné z: https://learning.lpi.org/en/learning-materials/101-500/ , 2. LPIC-1 Exam 102. LPI [online]. Canada: The Linux Professional Institute, 2021 [cit. 2021-9-22]. Dostupné z: https://learning.lpi.org/en/learning-materials/102-500/ , 3. Linux - Dokumentační projekt [online]. 4. Praha: Computer Press, 2007 [cit. 2021-9-22]. Dostupné z: https://i.iinfo.cz/files/root/k/LDP_4.pdf .					
Course language: Slovak or English					
Notes:					
Course assessment Total number of assessed students: 348					
A	B	C	D	E	FX
30.46	21.84	18.97	9.48	6.9	12.36

Provides: RNDr. Richard Staňa, PhD., Mgr. Samuel Baran, Mgr. Lukáš Zmuda

Date of last modification: 10.01.2026
--

Approved:

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice					
Faculty: Faculty of Science					
Course ID: ÚINF/ LAD1/15		Course name: Logical aspects of databases			
Course type, scope and the method: Course type: Lecture Recommended course-load (hours): Per week: 2 Per study period: 28 Course method: present					
Number of ECTS credits: 4					
Recommended semester/trimester of the course:					
Course level: II., N					
Prerequisites:					
Conditions for course completion: Satisfiable understanding of basic concepts.					
Learning outcomes: Ability to correctly formalize databases.					
Brief outline of the course: 1.-3. Basic concepts of logic – a symbol, a term, a formula, an interpretation 4. Formalization of a table and a database 5. Conjunctive queries 6. Conjunctive calculus 7. Relations between conjunctive calculus and conjunctive queries 8.-10. Relational algebra 11.-12. Relations of different models of databases					
Recommended literature: https://ics.upjs.sk/~krajci/skola/vyucba/ucebneTexty/LAD-presentation.pdf					
Course language: Slovak					
Notes:					
Course assessment Total number of assessed students: 101					
A	B	C	D	E	FX
47.52	17.82	15.84	9.9	6.93	1.98
Provides: prof. RNDr. Stanislav Krajči, PhD.					
Date of last modification: 23.11.2021					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ MTL/22	Course name: MATLAB and neurocognition
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 1 / 2 Per study period: 14 / 28 Course method: present	
Number of ECTS credits: 3	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites:	
Conditions for course completion: Written quizzes, midterm and final exam.	
Learning outcomes: Intro to programming in MATLAB with focus on its usage in neural and cognitive Science.	
Brief outline of the course: <ol style="list-style-type: none"> 1. Intro to Matlab 2. Navigation, interaction, variables, vectors, matrices, scripts, toolboxes 3. Interaction with humans in behavioral experiments 4. Auditory and visual stimulus generation 5. Analysis and visualization of behavioral data 6. Analysis of neurophysiological data 7. Analysis of neuroimaging data. 8. Cognitive and neural modeling in Matlab 9. Auditory modeling tools 10. Visual modeling tools 11. Tools for modeling of learning 12. Tools for psychological experiments 	
Recommended literature: <ol style="list-style-type: none"> 1. Wallisch P, et al. MATLAB for Neuroscientists: An Introduction to Scientific Computing in MATLAB. Academic Press 2008. ISBN-13: 978-0123838360 2. Stork D, Yom-Tov E: Computer Manual in MATLAB to accompany Pattern Classification, 2nd Edition, Wiley, 2004 ISBN-13: 978-0471429777 3. Dayan P and LF Abbott: Theoretical Neuroscience - Computational and Mathematical Modeling of Neural Systems. MIT Press, 2005 ISBN-13: 978-0262541855 	
Course language: Slovak or English	
Notes: Content prerequisites: basic programming skills or instructor's consent	

Course assessment					
Total number of assessed students: 16					
A	B	C	D	E	FX
6.25	25.0	50.0	18.75	0.0	0.0
Provides: doc. Ing. Norbert Kopčo, PhD., univerzitný profesor, Ing. Peter Lokša, PhD., RNDr. Keerthi Kumar Doreswamy, PhD., Ing. Udbhav Singhal, PhDr. Myroslav Fedorenko					
Date of last modification: 04.04.2022					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ PDS1/21	Course name: Parallel and distributed systems
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 2 / 2 Per study period: 28 / 28 Course method: present	
Number of ECTS credits: 5	
Recommended semester/trimester of the course:	
Course level: II., N	
Prerequisites:	
Conditions for course completion: Home assignments, class project from tutorials, midterm written exam. Final written and oral exam.	
Learning outcomes: Understand the principles, basic problems and algorithms of parallel programming. Be able to implement synchronization procedures and manage and use interprocess communication. Master the basics of GPU programming. Understand the differences between parallel and distributed computational models. Master basic distributed algorithms and know how to implement them. Understand the problems of creating a distributed system environment and know how to solve them. Be able to use distributed environments in practical applications.	
Brief outline of the course: Parallel architectures, parallel computational model, access to shared memory. Basic algorithms, scaling, optimality. Effective methods of parallel search and sorting. Working in a GPU environment. Distributed computational model, communication protocols, characteristics of distributed systems. Intercomputer communication, distributed synchronization algorithms, transactions, termination and deadlock detection. Consistency issues with distributed memory sharing. Distributed application environment. Reliable calculations in an environment with errors.	
Recommended literature: 1. J. JáJá: An Introduction to Parallel Algorithms, Addison-Wesley, 1992, ISBN 0-201-54856-9 2. P. Sanders, K. Mehlhorn, M. Dietzfelbinger, R. Dementiev: Sequential and Parallel Algorithms and Data Structures, Springer, 2019 3. Sukumar Ghosh: Distributed Systems and Algorithms (Second Edition), CRC Press 2014 4. M. Raynal: Distributed Algorithms for Message-Passing Systems, Springer, 2013 5. Gerard Tel: Introduction to Distributed Algorithms, Cambridge University Press, 2001	
Course language: Slovak or English	
Notes: Content prerequisites: basic of concurrent programming, basic of operating system principles	

Course assessment					
Total number of assessed students: 78					
A	B	C	D	E	FX
19.23	7.69	19.23	20.51	24.36	8.97
Provides: doc. RNDr. Jozef Jirásek, PhD., RNDr. Rastislav Krivoš-Belluš, PhD., RNDr. Ladislav Mikeš, PhD., doc. RNDr. Ľubomír Antoni, PhD.					
Date of last modification: 23.11.2021					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ SPP1b/22	Course name: Programming environments in schools II
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 2 / 2 Per study period: 28 / 28 Course method: present	
Number of ECTS credits: 4	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites: ÚINF/SPP1a/15	
Conditions for course completion: Conditions for ongoing evaluation: <ol style="list-style-type: none"> 1. Educational software or game programmed in the Scratch environment, 2. A programming etude created for learning of programming in the MIT App Inventor environment. 3. Educational or assistive software programmed in the MIT App Inventor environment. 4. A programmed project using the BBC micro: bit kit. Conditions for successful completion of the course: Obtaining at least 50% of points for ongoing assignments.	
Learning outcomes: After completing this course, students are able to: <ol style="list-style-type: none"> a) get an overview of educational programming environments, b) acquire programming skills in selected educational programming environments, c) develop the ability to design and program educational software for devices using their sensors and actuators. 	
Brief outline of the course: <ol style="list-style-type: none"> 1. Teaching algorithmization and programming in primary and secondary school - objectives, content, textbooks and methodological materials. Algorithmic computer games. 2. Programming in the Scratch environment. 3. Programming in the Scratch environment. 4. Programming in the Scratch environment. 5. Programming of mobile devices in the MIT App Inventor environment. 6. Programming of mobile devices in the MIT App Inventor environment. 7. Programming of mobile devices in the MIT App Inventor environment. 8. Programming of mobile devices in the MIT App Inventor environment. 9. Programming of mobile devices in the MIT App Inventor environment. 10. Programming BBC micro: bit kits in MS MakeCode environment. 11. Programming BBC micro: bit kits in MS MakeCode environment. 12. Overview of educational programming initiatives and development environments. 	
Recommended literature:	

BELL, Charles A., 2017. Micropython for the internet of things: a beginner's guide to programming with Python on microcontrollers. New York, NY: Springer Science+Business Media. ISBN 9781484231227.

GUTSCHANK, Jörg et al., 2019. Coding in STEM Education [online]. Berlin: Science on Stage Deutschland e.V., 76 p. [cited 2021-7-10]. ISBN 978-3-942524-58-2. Available from: https://www.science-on-stage.eu/sites/default/files/material/coding_in_stem_education_en_2nd_edition.pdf

ŠNAJDER, Ľubomír, Gabriela LOVÁSZOVÁ, Viera MICHALIČKOVÁ and Ján GUNIŠ, 2020. Programovanie mobilných zariadení [online]. Bratislava: Centrum vedecko-technických informácií SR, 300 p. [cited 2020-11-30]. ISBN 978-80-89965-63-2. Available from: <https://registracia.itakademia.sk/media/themes/nip-pmz.pdf>

WOLBER, David, 2014. App Inventor: Vytvořte si vlastní aplikaci pro Android. Brno: Computer Press. ISBN 978-80-251-4195-3.

LOVÁSZOVÁ, Gabriela, Jana GALBAVÁ, Viera PALMÁROVÁ and Monika TOMCSÁNYIOVÁ, 2010. Ďalšie vzdelávanie učiteľov základných škôl a stredných škôl v predmete informatika: Malé programovacie jazyky. Bratislava: Štátny pedagogický ústav. ISBN 978-80-8118-066-8.

CODE.ORG. Learn today, build a brighter tomorrow.
Code.org [online]. [cited 2021-7-13]. Available from: <https://code.org/>

THE LIFELONG KINDERGARTEN GROUP AT MIT MEDIA LAB. Scratch - Imagine, Program, Share [online]. [cited 2021-7-13]. Available from: <https://scratch.mit.edu/>

MASSACHUSETTS INSTITUTE OF TECHNOLOGY. MIT App Inventor
Explore MIT App Inventor [online]. [cited 2021-7-13]. Available from: <http://appinventor.mit.edu/>

MICRO:BIT EDUCATIONAL FOUNDATION. BBC micro:bit [online]. [cited 2021-7-13]. Available from: <https://microbit.org/>

SPY O.Z. Učíme s Hardvérom [online]. [cited 2021-7-13]. Available from: <https://www.ucimeshardverom.sk/>

Course language:

Slovak or English

Notes:

By default, teaching is carried out face to face. If this is not possible (eg due to a pandemic), teaching is provided at a distance through video conferencing programs and LMS.

Course assessment

Total number of assessed students: 47

A	B	C	D	E	FX
42.55	19.15	10.64	17.02	2.13	8.51

Provides: doc. RNDr. Ľubomír Šnajder, PhD., RNDr. Katarína Brinziková

Date of last modification: 08.02.2022

Approved:

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ PJP/25	Course name: Programming language Python
Course type, scope and the method: Course type: Lecture / Practice Recommended course-load (hours): Per week: 1 / 2 Per study period: 14 / 28 Course method: present	
Number of ECTS credits: 4	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites: ÚINF/PAZ1a/15	
Conditions for course completion: At least 50 % of the marks in the continuous assessment A minimum of 50 % marks in the mid-term test and the final exam practical test	
Learning outcomes: Implement solutions to selected problems in Python using available modules. Use and implement non-trivial algorithms to solve selected problems. Use an object-oriented approach to problem solving. Program in Python in an object-oriented manner using Python specifics. Test programs. Implement parallel computing.	
Brief outline of the course: 1. Introduction to the environment, basic features of Python, simple and structured data types. 2. Input, output, function definition, lambda function, generator notation, function as parameter, string formatting. 3. Control structures, iterating over data structures, context manager. 4. Exception handling and exception raising. Philosophy of exceptions in Python. 5. Working with files. Serialization and deserialization of data - json and pickle protocol. Text and binary files. Manipulation with files. Open data. 6. Object-oriented programming 1. Design of custom classes, special methods, properties, philosophy of accessing methods and attributes. 7. Object-oriented programming 2. Comparison and differences with Java. Multiple inheritance. 8. Method overloading. Static methods, abstract classes, data class. 9. Decorators, memoization, modules, packages. 10. Code validation (debugging), testing (doctest, unittest), test-driven development. 11. Parallel computing, processes, process triggering and inter-process communication (shared variable, pipe, queue). 12. Graphical program design and implementation.	
Recommended literature: PILGRIM, Mark, 2011. Ponořme se do Pythonu(u) 3 [online]. Praha: CZ.NIC, z. s. p. o. ISBN 978-80-904248-2-1. Available at: https://www.nic.cz/files/edice/python_3.pdf	

<p>PIRNAT, Mike, 2015. How to Make Mistakes in Python [online]. Boston: O'Reilly Media. ISBN 978-1-4919-3447-0. Available at: https://www.dbooks.org/how-to-make-mistakes-in-python-1491934476/</p> <p>STACK OVERFLOW CONTRIBUTORS, 2018. Python® Notes for Professionals [online]. B.m.: GoalKicker. Available at: https://books.goalkicker.com/PythonBook/PythonNotesForProfessionals.pdf</p> <p>ROSEMAN, Mark, 2024. Modern Tk Best Practices [online]. 2024. Available at: https://tkdocs.com/</p>					
<p>Course language: Slovak language, knowledge of English language is only required to read documentation of Python.</p>					
<p>Notes:</p>					
<p>Course assessment Total number of assessed students: 41</p>					
A	B	C	D	E	FX
4.88	12.2	41.46	24.39	7.32	9.76
<p>Provides: PaedDr. Ján Guniš, PhD., univerzitný docent, RNDr. Zoltán Szoplák, doc. RNDr. Ľubomír Šnajder, PhD., Ing. Peter Lokša, PhD., Mgr. Samuel Baran, Mgr. Norbert Michel</p>					
<p>Date of last modification: 08.03.2025</p>					
<p>Approved:</p>					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ PRO1b/25	Course name: Project II.
Course type, scope and the method: Course type: Practice Recommended course-load (hours): Per week: Per study period: 52s Course method: present	
Number of ECTS credits: 4	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites:	
Conditions for course completion: Active participation in the project. Participating in regular project team meetings. Presentation of the results achieved in solving a specific problem. Uploading a software work. Preparation of materials for the promotion of the final work.	
Learning outcomes: Learn how to work on a larger software part at all stages of its life cycle. Be able to analyze and explicitly express user requirements, precisely specify the task, design a solution and evaluate alternatives. Implement and test an effective and correctly designed solution. Learn to keep detailed documentation and present the results of the work in writing and in public. Learn to work together in a development team, share work effectively and exchange ideas.	
Brief outline of the course: The course is realized as part of "Živé projekty" (Live projects) in cooperation with the Technical University of Košice and several software companies. Students work in a team of 4-5 members to develop, test and present a software product under the guidance of a mentor from a university or a software company. <ol style="list-style-type: none"> 1. Team creation and project selection takes place at the beginning of October 2. Students meet with the project mentor on a weekly basis and continuously work on the creation of a software product 3. Around mid-January, students submit a video with a short presentation of the project 4. At the beginning of February, the project presentation takes place. The best teams are awarded with material prizes. 	
Recommended literature: The sources of information depend on the selected project.	
Course language: Slovak or english	
Notes: Content prerequisites: advanced programming skills	

Course assessment					
Total number of assessed students: 113					
A	B	C	D	E	FX
62.83	15.04	8.85	5.31	2.65	5.31
Provides: RNDr. Peter Gurský, PhD., univerzitný docent					
Date of last modification: 08.04.2025					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ MSW/25	Course name: Software Systems Modelling
Course type, scope and the method: Course type: Practice Recommended course-load (hours): Per week: 3 Per study period: 42 Course method: present	
Number of ECTS credits: 4	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites: ÚINF/PAZ1b/15	
Conditions for course completion: The course grade is based on the evaluation of partial tasks within the semester project and assigned papers. Within the individual tasks, students must demonstrate their ability to be oriented in the subject matter, mastery of the theoretical foundations of modeling, and basic skills for creating and interpreting software models. The assessment is awarded if the student obtains at least 50% of the possible scores. Detailed requirements are listed in AIS.	
Learning outcomes: By completing this course, students will: <ul style="list-style-type: none"> - gain knowledge of the theoretical principles and fundamentals of SW system modeling, particularly using UML and BPMN, - master the basic principles of creating dynamic and statistical models of software systems, - become familiar with standard languages for modeling software systems, - gain practical experience in creating models using selected modeling tools, - gain knowledge of other selected areas of software engineering. 	
Brief outline of the course: <ol style="list-style-type: none"> 1. Evolution of software system architecture. Task analysis and formalization of inputs for SW systems. 2. Model, metamodel, modeling language. Transformation between models. 3. Requirements gathering and system analysis. Use case diagram. Activity diagram. Process model. 4.-5. Petri nets. 6. Orchestration in process modeling. 7. Choreography in process modeling. 8. Class diagram and object diagram. 9. Sequence diagram and communication diagram. 10. State diagram and advanced process modeling capabilities. 11. Architecture and deployment. Component diagram and deployment diagram. 12. Testing and documentation. Package diagram. 13. Model-driven development (MDA) and model-driven architecture (MDA). Process automation. 	

Recommended literature:

1. I. Sommerville, Software Engineering, 10th ed. London: Pearson, 2015.
2. S. Cook et al., Unified Modeling Language (UML) Version 2.5.1. Standard. Object Management Group (OMG), December 2017. Available online: <https://www.omg.org/spec/UML/2.5.1>
3. Free Online Courses and Resources Every Software Engineer Needs. Available online: <https://www.indeed.com/career-advice/news/software-engineer-free-online-courses-resources>.
4. Weske, M. Business Process Management, Springer 2007
5. White S.A., Miers D., Fischer L., BPMN Modeling and Reference Guide, Future Strategies Inc., Lighthouse Pt 2008
6. White:, S.A. Process Modeling Notations and Workflow Patterns, Dostupné online http://www.omg.org/bp-#corner/bp-#files/Process_Modeling_Notations.pdf.
7. Object Management Group. Business Process Model and Notation (BPMN), Version 2.0.2. December 2013. Available online: <https://www.omg.org/spec/BPMN/2.0.2/PDF/>

Course language:

Slovak and/or English language.

Notes:

Content prerequisites:
programming, fundamentals of software engineering, fundamentals of database systems,
fundamentals of project management

Course assessment

Total number of assessed students: 250

A	B	C	D	E	FX
48.4	22.8	14.8	7.2	4.0	2.8

Provides: prof. RNDr. Gabriel Semanišin, PhD.

Date of last modification: 15.09.2025

Approved:

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ TSD/19	Course name: Technologies of big data processing
Course type, scope and the method: Course type: Practice Recommended course-load (hours): Per week: 2 Per study period: 28 Course method: present	
Number of ECTS credits: 2	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites:	
Conditions for course completion: Active participation, written test, class project.	
Learning outcomes: Practical experience with modern Big Data processing and storage systems. Introduction to their architecture and implementation.	
Brief outline of the course: 1. Introduction to Big Data processing. Freely accessible datasets. 2. Cloud environment. 3. Distributed file systems, object storage. Data formats. 4. Scalability, hashing, data sharding. 5. Distributed databases, consistency trade-offs. NoSQL. 6. Batch data processing: MapReduce 7. Batch data processing: Spark I 8. Batch data processing: Spark II 9. Stream data processing: Kafka 10. Stream data processing: Beam 11. Distributed neural network training.	
Recommended literature: 1. KLEPPMANN, Martin. Designing data-intensive applications: the big ideas behind reliable, scalable, and maintainable systems. Beijing: O'Reilly, 2017. ISBN 978-1-449-37332-0. 2. WHITE, Tom. Hadoop: the definitive guide. 3rd ed. Sebastopol: O'Reilly, 2012. ISBN 978-1-449-31152-0. 3. MARZ, Nathan a James WARREN. Big data: principles and best practices of scalable real-time data systems. Shelter Island, NY: Manning, [2015]. ISBN 978-1-617290-34-3. 4. PENTREATH, Nick. Machine Learning with Spark; Packt Publishing, [2015]. ISBN 978-1-783288-51-9.	
Course language: Slovak or English	
Notes:	

Content prerequisites: database basics, Python programming					
Course assessment					
Total number of assessed students: 30					
A	B	C	D	E	FX
93.33	6.67	0.0	0.0	0.0	0.0
Provides: Bc. Marián Dvorský, doc. RNDr. Ľubomír Antoni, PhD.					
Date of last modification: 04.01.2022					
Approved:					

COURSE INFORMATION LETTER

University: P. J. Šafárik University in Košice	
Faculty: Faculty of Science	
Course ID: ÚINF/ TYS1/15	Course name: Typographical systems
Course type, scope and the method: Course type: Practice Recommended course-load (hours): Per week: 2 Per study period: 28 Course method: present	
Number of ECTS credits: 2	
Recommended semester/trimester of the course:	
Course level: I., N	
Prerequisites:	
Conditions for course completion: Satisfiable ability to correct mainly mathematical typesetting.	
Learning outcomes: To provide the basic information on principles for typesetting of documents containing mathematical formulas.	
Brief outline of the course: <ol style="list-style-type: none"> 1. Principles for typesetting of documents containing mathematical formulas. 2. Typesetting of a plain text, special text symbols, using of text fonts.3 3. TeX macros. 4. Enumerations in text and footnote command. Parameter setting determining the appearance of the pages. 5. Typesetting of mathematical formulas in text and displays, aligning formulas. 6. Making tables and pictures. 7. Definitions, theorems, and proofs in a mathematical document. 8. Contents, bibliography, sections in a document. 9. Pictures. 10.-12. Project. 	
Recommended literature: <ol style="list-style-type: none"> 1. D. E. Knuth, The TeXbook, Computers and Typesetting, Addison-Wesley, Reading, Massachusetts, 1986. 2. M. Doob, Jemný úvod do TeXu, CSTUG, 1990; český překlad z "A Gentle Introduction to TeX" (text vo¾ane prístupný v CTAN archíve). 3. O. Ulrych, AMS-TeX za 59 minút, (verzia 1.0), Praha, 1989. 4. J. Chlebíková, AMS-TeX (verzia 2.0), Bratislava, 1992. 5. M. Spivak, The Joy of TeX, Amer. Math. Soc., 1986. 6. L. Lamport, LaTeX: A Document Preparation System, Addison-Wesley, Massachusetts, 1986. 7. L. Lamport, MakeIndex: An index processor for LaTeX, 17 February 1987. 8. J. Rybička, LaTeX pro začátečníky, Konvoj, Brno, 1995. 9. H. Partl, E. Schlegl, I. Hyna, P. Sýkora, LaTeX – Stručný popis. 	

10. T. Oetiker, H. Partl, I. Hyna, E. Schlegl, M. Kocer, P. Sýkora, Ne příliš stručný úvod do systému LaTeX2e (neboli LaTeX2e v 73 minutách).
11. M. Goossens, F. Mittelbach, and A. Samarin, The LaTeX Companion, Addison-Wesley, Reading, Massachusetts, 1994. Kapitola 8 je volne prístupná v TeX archívoch (ch8.pdf). 4
12. G. Grätzer, Math into LaTeX, 3rd edition, Birkhäuser, Boston, 2000.

Course language:

Slovak.

Notes:

Course assessment

Total number of assessed students: 267

A	B	C	D	E	FX
50.56	16.85	19.48	5.99	6.37	0.75

Provides: prof. RNDr. Stanislav Krajči, PhD.

Date of last modification: 08.01.2022

Approved: