Analysis of Algorithms

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of study : full-time studies

Study profile : general academic

Mandatory status : compulsory

Education cycle : 2022/23

Course code : UJ.WMIIANS.1100.03358.22

Language of instruction : Polish

Course related to scientific research : Yes

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.AA1.OL

Course coordinator

Maciej Ślusarek

Course instructor

Maciej Ślusarek

Period Semester 5	Form of verification of learning outcomes exam Teaching format and hours lecture: 30 tutorials: 30	Number of ECTS credits 6.0

Learning outcomes for the course

Code	Outcomes in terms of	Directional learning outcomes	Verification methods

Code	Outcomes in terms of	Directional learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	knows basic methods of probabilistic algorithm analysis and can apply them in selected areas of algorithmics	IAN_K1_W06, IAN_K1_W11	oral exam, credit
W2	knows the amortized analysis method and can use it to analyze a sequence of operations on a data structure	IAN_K1_W06, IAN_K1_W11	oral exam, credit
W3	knows selected advanced algorithms and data structures for problems related to sorting and searching and can analyze their complexity	IAN_K1_W06, IAN_K1_W11	oral exam, credit
Skills – The student can:			
U1	uses algorithm analysis to assess the possibility of effectively solving a given problem and to estimate the effectiveness of a given solution	IAN_K1_U01, IAN_K1_U02, IAN_K1_U06, IAN_K1_U10, IAN_K1_U11, IAN_K1_U17, IAN_K1_U21	oral exam, credit
Social competences – The student is ready to:			
K1	precisely formulate questions for the analysis of a given algorithmic problem	IAN_K1_K01	oral exam, credit

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types	
lecture	30	
tutorials	30	
preparation for tutorials	60	
exam preparation	40	
Total student workload	Number of hours 160	ECTS 6.0

* hour (lesson) means 45 minutes

Course content

No.	Program content	Learning outcomes for the course
1.	1. Recurrence equations in algorithm analysis. Master theorem, variants. 2. Elements of probability calculus: indicator random variables, secretary problem, generating random permutations. 3. Generating functions technique in average case analysis, example: analysis of the secretary problem. 4. Analysis of the Quicksort algorithm: expected value and variance of the number of comparisons. 5. Lower bounds on sorting complexity, Ford-Johnson algorithm and the problem of minimum number of comparisons. 6. Probabilistic analysis of tree-based dictionary implementations - binary search trees and treaps. 7. Amortized analysis using the example of splay trees and the static dictionary problem. 8. Find-Union problem, analysis using iterated logarithm. 9. Interpolation search, quadratic method and its complexity. 10. Hashing: open hashing analysis, universal families of hash functions, perfect hashing. 11. Selected online algorithms: graph coloring, maximum matching in bipartite graphs. 12. Priority queues and mergeable heaps: Fibonacci heaps, van Emde Boas queues, applications in graph algorithms.	W1, W2, W3, U1, K1

Extended information

Teaching methods:

conventional lecture, subject tutorials

Class type	Credit forms	Course credit conditions
lecture	oral exam	Positive exam grade. Admission to the exam provided positive grade from tutorials. The final grade is the average of the tutorial grade and the exam.
tutorials	credit	Credit based on homework assignments and tests.

Prerequisites and additional requirements

Algorithms and Data Structures 2, Discrete Mathematics, Probabilistic Methods in Computer Science

Literature

Required

1. T.H.Cormen, Ch.E.Leiserson, R.L.Rivest, C. Stein, Wprowadzenie do algorytmów, wydanie III, WNT, 2012

Additional

1. L.Banachowski, K.Diks, W.Rytter, Algorytmy i struktury danych, WNT, 2001
2. L.Banachowski, A.Kreczmar, W.Rytter, Analiza algorytmów i struktur danych, WNT, 1987

3. A.V.Aho, J.E.Hopcroft, J.D.Ullman, Projektowanie i analiza algorytmów, PWN 1985, Helion 2003

Algebraic and Number Theory Algorithms

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of study : full-time studies

Study profile : general academic

Mandatory status : optional

Education cycle : 2022/23

Course code : UJ.WMIIANS.1380.03348.22

Language of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.AATL.S

Course coordinator

Lech Duraj

Course instructor

Lech Duraj

Periods Semester 4, Semester 5, Semester 6	Form of verification of learning outcomes exam Teaching format and hours lecture: 30 tutorials: 30	Number of ECTS credits 6.0

Educational goals for the course

C1	The aim of the course is to provide knowledge in the field of algorithms related to number theory and algebra, primarily in relation to cryptography as their main application.

Learning outcomes for the course

Code	Outcomes in terms of	Directional learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	basic concepts in the field of algebra and number theory listed in "Course content", useful in the work of a computer scientist	IAN_K1_W01	oral exam, credit
W2	algorithms listed in "Course content" (in particular cryptographic algorithms)	IAN_K1_W06, IAN_K1_W08, IAN_K1_W10	oral exam, credit
Skills – The student can:			
U1	conduct proofs of correctness of selected theorems given in "Course content", in particular proofs of correctness and complexity analysis of algorithms in the field of algebra and number theory	IAN_K1_U01, IAN_K1_U10, IAN_K1_U17, IAN_K1_U21	oral exam, credit
U2	implement basic algorithms of algebra and number theory (including cryptographic algorithms) in an efficient way, taking into account communication security issues	IAN_K1_U03, IAN_K1_U17	credit

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types
lecture	30
tutorials	30
solving computer tasks independently	30
preparation for tutorials	30
solving problem tasks	30
exam preparation	29
exam participation	1

Total student workload	Number of hours 180	ECTS 6.0

* hour (lesson) means 45 minutes

Course content

No.	Program content	Learning outcomes for the course
1.	Integers: computer representation, basic arithmetic algorithms (multiplication, division with remainder, calculating the greatest common divisor), complexity of sorting and searching on integers	W1, W2, U1, U2
2.	Algebraic constructions: commutative groups, rings and fields, non-commutative groups (permutations), polynomials and finite fields (including arithmetic operations), elliptic curve arithmetic	W1, W2, U1, U2
3.	Basics of cryptography: symmetric algorithms, public key cryptography, RSA algorithm, Diffie-Hellman protocol, ElGamal algorithm	W1, W2, U1, U2
4.	Prime numbers and factorization: Miller-Rabin probabilistic test, outline of the AKS deterministic test, Pollard's "rho" algorithm, quadratic sieve, number field sieve	W1, W2, U1, U2
5.	Discrete root problem and discrete logarithm problem on integers and in commutative groups (Tonelli-Shanks algorithm, baby-step-giant-step method, Pohlig-Hellman algorithm, index calculus)	W1, W2, U1, U2
6.	Basics of quantum computing, Shor's algorithm	W1, W2, U1, U2

Extended information

Teaching methods:

multimedia presentation lecture, discussion, problem solving, subject tutorials

Class type	Credit forms	Course credit conditions
lecture	oral exam	Positive exam grade and combined positive grade from exam and tutorials
tutorials	credit	Credit for tutorials based on assignment programs and homework

Literature

Required

1. Neal Koblitz, "A Course in Number Theory and Cryptography"

Additional

1. Song Y. Yan, "Number Theory for Computing"

Calculus

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of study : full-time studies

Study profile : general academic

Mandatory status : compulsory

Education cycle : 2022/23

Course code : UJ.WMIIANS.110.02906.22

Language of instruction : Polish

Disciplines : Mathematics

ISCED classification : 0541 Mathematics

USOS code : WMI.TCS.AM1.OL

Course coordinator

Rafał Pierzchała

Course instructor

Rafał Pierzchała

Period Semester 1	Form of verification of learning outcomes graded credit Teaching format and hours lecture: 30 tutorials: 30	Number of ECTS credits 5.0

Learning outcomes for the course

Code	Outcomes in terms of	Directional learning outcomes	Verification methods

Code	Outcomes in terms of	Directional learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	theorems covered in the lecture, listed in the "Course content" field	IAN_K1_W01, IAN_K1_W12	graded credit
Skills – The student can:			
U1	provide examples of applications of theorems learned during the lecture and solve typical problems related to these theorems	IAN_K1_U02	graded credit
Social competences – The student is ready to:			
K1	precisely formulate questions for the analysis of a given topic	IAN_K1_K01	no credit

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types	
lecture	30	
tutorials	30	
preparation for classes	90	
Total student workload	Number of hours 150	ECTS 5.0

* hour (lesson) means 45 minutes

Course content

No.	Program content	Learning outcomes for the course

No.	Program content	Learning outcomes for the course
1.	Introductory information. Sequences and series. Properties of convergent sequences. Numerical series. Convergence criteria for series. Limits and continuity. Function limit. One-sided limits. Theorems on the relationship between limits and operations. Continuous functions and their properties. Differential calculus of functions of one variable. Definition of derivative. Theorems on differentiation of sum, product, quotient, composition, and inverse function. Rolle's and Lagrange's theorems. L'Hospital's rules. Derivatives of orders higher than 1. Taylor's formula. Investigating function properties. Integral calculus of functions of one variable. Theorems on integration by parts and integration by substitution.	W1, U1, K1

Extended information

Teaching methods:

conventional lecture, subject tutorials

Class type	Credit forms	Course credit conditions
lecture	no credit	Positive grade from tutorials.
tutorials	graded credit	Oral or written tests. Problems to solve independently. Activity during classes.

Prerequisites and additional requirements

attendance at tutorials is mandatory

Literature

Required

1. F. Leja, Rachunek różniczkowy i całkowy, Państwowe Wydawnictwo Naukowe, Warszawa 1969.
2. W. Rudin, Podstawy analizy matematycznej, Państwowe Wydawnictwo Naukowe, Warszawa 1982.
3. G.M. Fichtenholz, Rachunek różniczkowy i całkowy (tomy I i II), PWN Warszawa 1995.
4. W. Krysicki, L. Włodarski, Analiza matematyczna w zadaniach, część I, Wydawnictwo Naukowe PWN, Warszawa 2006.

Calculus 2

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of study : full-time studies

Study profile : general academic

Mandatory status : compulsory

Education cycle : 2022/23

Course code : UJ.WMIIANS.120.02909.22

Language of instruction : Polish

Disciplines : Mathematics

ISCED classification : 0541 Mathematics

USOS code : WMI.TCS.AM2.OL

Course coordinator

Rafał Pierzchała

Course instructor

Rafał Pierzchała

Period Semester 2	Form of verification of learning outcomes exam Teaching format and hours lecture: 30 tutorials: 30	Number of ECTS credits 6.0

Learning outcomes for the course

Code	Outcomes in terms of	Directional learning outcomes	Verification methods

Code	Outcomes in terms of	Directional learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	theorems covered in the lecture, listed in the "Course content" field	IAN_K1_W01, IAN_K1_W12	written exam, graded credit
Skills – The student can:			
U1	provide examples of applications of theorems learned during the lecture and solve typical problems related to these theorems	IAN_K1_U02	written exam, graded credit
Social competences – The student is ready to:			
K1	precisely formulate questions for the analysis of a given topic	IAN_K1_K01	graded credit

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types	
lecture	30	
tutorials	30	
preparation for tutorials	90	
exam preparation	28	
exam participation	2	
Total student workload	Number of hours 180	ECTS 6.0

* hour (lesson) means 45 minutes

Course content

No.	Program content	Learning outcomes for the course
1.	Basic topological concepts. Limits and continuity of multivariate functions. Function sequences and series. Theorems on continuity, integrability, and differentiability of function sequence limits. Power series. Partial derivatives. Differentiability. Taylor's formula. Extrema of multivariate functions. Implicit function theorem. Local diffeomorphism theorem. Integration of multivariate functions. Fubini's theorem. Change of variables theorem.	W1, U1, K1

Extended information

Teaching methods:

conventional lecture, subject tutorials

Class type	Credit forms	Course credit conditions
lecture	written exam	Positive exam grade.
tutorials	graded credit	Oral or written tests. Problems to solve independently. Activity during classes.

Prerequisites and additional requirements

Completed Calculus course. Attendance at tutorials is mandatory.

Literature

Required

1. F. Leja, Rachunek różniczkowy i całkowy, Państwowe Wydawnictwo Naukowe, Warszawa 1969.
2. W. Rudin, Podstawy analizy matematycznej, Państwowe Wydawnictwo Naukowe, Warszawa 1982.
3. G.M. Fichtenholz, Rachunek różniczkowy i całkowy (tomy I, II i III), PWN Warszawa 1995.
4. W. Krysicki, L. Włodarski, Analiza matematyczna w zadaniach, część I i II, Wydawnictwo Naukowe PWN, Warszawa 2006.

Numerical Algorithms

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of study : full-time studies

Study profile : general academic

Mandatory status : optional

Education cycle : 2022/23

Course code : UJ.WMIIANS.1380.03349.22

Language of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.AN.S

Course coordinator

Lech Duraj

Course instructor

Lech Duraj

Periods Semester 4, Semester 5, Semester 6	Form of verification of learning outcomes exam Teaching format and hours lecture: 30 laboratory exercises: 30	Number of ECTS credits 6.0

Educational goals for the course

C1	The aim of the course is to provide knowledge in the field of numerical algorithms, with particular emphasis on practically applied and experimentally verified algorithms, as well as analysis of algorithms in terms of numerical stability

Learning outcomes for the course

Code	Outcomes in terms of	Directional learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	issues listed in "Course content" regarding computer arithmetic, calculation errors, conditioning and numerical stability of algorithms	IAN_K1_W12	written exam, credit
W2	issues listed in "Course content" in the field of algebra and numerical analysis, including methods for solving numerical problems	IAN_K1_W09, IAN_K1_W10, IAN_K1_W12	written exam, credit
Skills – The student can:			
U1	solve numerical problems listed in "Course content", and effectively implement selected algorithms	IAN_K1_U01, IAN_K1_U05, IAN_K1_U10	written exam, credit
U2	prove correctness and numerical stability of algorithms, select appropriate algorithms to solve numerical problems	IAN_K1_U01, IAN_K1_U02, IAN_K1_U10, IAN_K1_U11, IAN_K1_U17	written exam, credit

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types
lecture	30
laboratory exercises	30
preparation for exercises	42
solving computer tasks independently	30

exam preparation	45
exam participation	3

Total student workload	Number of hours 180	ECTS 6.0

* hour (lesson) means 45 minutes

Course content

No.	Program content	Learning outcomes for the course
1.	Floating-point arithmetic, calculation and rounding errors, numerical conditioning and numerical stability of algorithms	W1, U2
2.	Numerical methods in algebra: systems of linear equations, Gaussian elimination, orthonormalization, LU and QR matrix decomposition, eigenvectors and eigenvalues, singular values and SVD decomposition	W2, U1, U2
3.	Numerical analysis: nonlinear methods, unconstrained and constrained optimization, interpolation and approximation, numerical differentiation and integration, fundamentals of differential equations, Fourier transform and related transformations	W2, U1, U2

Extended information

Teaching methods:

multimedia presentation lecture, discussion, problem solving

Class type	Credit forms	Course credit conditions
lecture	written exam	Positive combined grade from exam and exercises
laboratory exercises	credit	Solving an appropriate number of programming and whiteboard tasks

Literature

Required

1. Justin Solomon, "Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics"

Additional

1. David Bau III, Lloyd N. Trefethen, "Numerical Linear Algebra"

2. David Ronald Kincaid, Elliott Ward Cheney, "Numerical Analysis: Mathematics of Scientific Computing"

Algorithms and Data Structures 1

Course description

Basic Information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of studies : full-time studies

Study profile : general academic

Required : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.140.03340.22

Languages : Polish

Disciplines : Computer Science

ISCED Classification : 0613 Software and applications development and analysis

USOS Code : WMI.TCS.ASD1.OL

Course coordinator

Maciej Ślusarek

Course instructors

Maciej Ślusarek, Piotr Micek, Krzysztof Potępa

Period Semester 3	Form of verification of learning outcomes graded credit Course format and hours lecture: 30 laboratory classes: 30	ECTS credits 5.0

Learning outcomes for the course

Code	Outcomes in the area of:	Directional learning outcomes	Verification methods

Code	Outcomes in the area of:	Directional learning outcomes	Verification methods
Knowledge – Student knows and understands:			
W1	knows advanced data structures based on binary search trees: AVL trees, red-black trees, B-trees, heaps, splay trees and methods of their implementation	IAN_K1_W04, IAN_K1_W06, IAN_K1_W07, IAN_K1_W08, IAN_K1_W11	graded credit, credit
W2	has in-depth knowledge of algorithm construction techniques, in particular dynamic programming and the greedy method	IAN_K1_W06, IAN_K1_W07, IAN_K1_W08, IAN_K1_W09, IAN_K1_W10, IAN_K1_W11, IAN_K1_W12	graded credit, credit
W3	knows basic as well as selected advanced algorithms for many graph problems	IAN_K1_W06, IAN_K1_W07, IAN_K1_W09, IAN_K1_W10, IAN_K1_W11	graded credit, credit
Skills – Student can:			
U1	model problems presented in natural language using mathematical language and algorithmic concepts	IAN_K1_U01, IAN_K1_U03, IAN_K1_U05, IAN_K1_U06, IAN_K1_U07, IAN_K1_U08, IAN_K1_U10, IAN_K1_U11, IAN_K1_U17, IAN_K1_U21, IAN_K1_U22	graded credit
U2	design and implement algorithms using basic and selected advanced algorithmic techniques	IAN_K1_U06, IAN_K1_U07, IAN_K1_U08, IAN_K1_U10, IAN_K1_U11, IAN_K1_U17, IAN_K1_U21, IAN_K1_U22	graded credit
U3	test their program, look for errors and optimize it	IAN_K1_U05, IAN_K1_U11, IAN_K1_U18	graded credit
Social competences – Student is ready to:			
K1	formulate precise questions that serve to deepen or supplement their understanding of a given topic	IAN_K1_K01	graded credit

ECTS credit balance

Student's activity form	Average number of hours* allocated to completed activity types
lecture	30
laboratory classes	30
independently solving computer problems	60
preparation for classes	30

Total student workload	Hours 150	ECTS 5.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	1. Dynamic programming: DAG of subtasks, solution reconstruction, memory size problem. Examples: traveling salesman problem, knapsack problem, longest common subsequence and Hirschberg's algorithm, optimal BST trees. 2. Greedy algorithms - selected examples: Huffman codes, scheduling with delay minimization, optimal buffering in cache memory. 3. Balanced trees: AVL trees, red-black trees, B-trees. 4. Other tree balancing mechanisms: probabilistic (heaps), amortized (splay trees). 5. Connectivity problems in graphs, strongly connected components, biconnected components. 6. Shortest paths in graphs, algorithms: Bellman/Ford, Dijkstra, Warshall/Floyd, Johnson. 7. Minimum spanning trees, algorithms: Jarnik/Prim, Boruvka/Sollin, Kruskal; disjoint-set data structure. 8. Network flows, algorithms: Ford/Fulkerson, Edmonds/Karp, push-relabel. 9. Bipartite graph matching, "turbo matching" algorithm, Hopcroft/Karp algorithm.	W1, W2, W3, U1, U2, U3, K1

Extended information

Teaching methods:

conventional lecture, laboratory classes

Class type	Assessment forms	Course completion conditions
lecture	credit	Participation in lectures.
laboratory classes	graded credit	Laboratory completion based on assessment programs, homework assignments and tests.

Prerequisites and additional requirements

Programming Methods

Literature

Required

1. T.H. Cormen, Ch.E. Leiserson, R.L. Rivest, C. Stein, Wprowadzenie do algorytmów, wydanie III, PWN, 2012

Additional

1. L.Banachowski, K.Diks, W.Rytter, Algorytmy i struktury danych, PWN, 2018

Algorithms and Data Structures 2

Course description

Basic Information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of studies : full-time studies

Study profile : general academic

Required : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.180.03345.22

Languages : Polish

Course related to scientific research : Yes

Disciplines : Computer Science

ISCED Classification : 0613 Software and applications development and analysis

USOS Code : WMI.TCS.ASD2.OL

Course coordinator

Maciej Ślusarek

Course instructors

Maciej Ślusarek, Marcin Briański, Krzysztof Potępa

Period Semester 4	Form of verification of learning outcomes exam Course format and hours lecture: 30 laboratory classes: 30	ECTS credits 6.0

Learning outcomes for the course

Code	Outcomes in the area of:	Directional learning outcomes	Verification methods

Code	Outcomes in the area of:	Directional learning outcomes	Verification methods
Knowledge – Student knows and understands:			
W1	knows standard algorithms and data structures used in solutions to algorithmic problems in computational geometry, text processing, and number theory	IAN_K1_W04, IAN_K1_W06, IAN_K1_W07, IAN_K1_W08, IAN_K1_W09, IAN_K1_W10, IAN_K1_W11, IAN_K1_W12	oral exam, credit
W2	understands the concepts of computational complexity, knows the definition of the NP class and NP-complete problems, identifies example NP-complete problems, knows selected approximation algorithms	IAN_K1_W11	oral exam, credit
Skills – Student can:			
U1	model problems presented in natural language using mathematical language and advanced algorithmic concepts	IAN_K1_U01, IAN_K1_U06, IAN_K1_U07, IAN_K1_U17, IAN_K1_U21, IAN_K1_U22	oral exam, credit
U2	propose a solution for a typical algorithmic problem in the discussed fields and present its solution orally and in writing	IAN_K1_U03, IAN_K1_U06, IAN_K1_U10, IAN_K1_U11, IAN_K1_U17, IAN_K1_U21, IAN_K1_U22	oral exam, credit
U3	design and implement algorithms using basic and selected advanced algorithmic techniques	IAN_K1_U06, IAN_K1_U07, IAN_K1_U08, IAN_K1_U11, IAN_K1_U17	oral exam, credit
U4	has an in-depth ability to test their program, search for errors and optimize it	IAN_K1_U03, IAN_K1_U05	oral exam, credit

Code	Outcomes in the area of:	Directional learning outcomes	Verification methods
Social competences – Student is ready to:			
K1	formulate precise questions that serve to deepen or supplement their understanding of a given topic	IAN_K1_K01	oral exam, credit

ECTS credit balance

Student's activity form	Average number of hours* allocated to completed activity types	
lecture	30	
laboratory classes	30	
independently solving computer problems	60	
preparation for classes	30	
preparation for exam	30	
Total student workload	Hours 180	ECTS 6.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course

No.	Course content	Learning outcomes for the course
1.	1. Pattern searching in text: prefix-suffixes, KMP method, Aho-Corasick automaton, Karp-Rabin algorithm, Karp-Miller-Rosenberg algorithm. 2. Suffix arrays: construction algorithms, longest common prefix array and optimal search algorithm, suffix trees and their relationship with suffix arrays. 3. Basic techniques of computational geometry: vector determinant, polar sorting, sweep line, convex hull algorithms, finding intersections of segments. 4. Further geometric algorithms: point-in-polygon test, representation of plane division, point location on a plane using layer method, kd-trees. 5. Linear programming, simplex method, duality. 6. Number theory problems: Euclidean algorithm, modular arithmetic, discrete logarithm, RSA algorithm. 7. Prime numbers, Miller-Rabin algorithm. 8. Fast Fourier Transform. 9. Computational complexity: NP class, NP-complete problems, examples of NP-completeness proofs, approximation algorithms.	W1, W2, U1, U2, U3, U4, K1

Extended information

Teaching methods:

conventional lecture, laboratory classes

Class type	Assessment forms	Course completion conditions
lecture	oral exam	Positive grade from the exam, covering the scope of ASD1 and ASD2 courses. Admission to the exam is conditional on a positive grade from the laboratory. The final grade is a weighted average of grades from ASD1 and ASD2 laboratories and the exam.
laboratory classes	credit	Laboratory completion based on assessment programs, homework assignments and tests.

Prerequisites and additional requirements

Algorithms and Data Structures 1, Discrete Mathematics

Literature

Required

1. T.H. Cormen, Ch.E. Leiserson, R.L. Rivest, C. Stein, Wprowadzenie do algorytmów, wydanie III, PWN, 2012

Additional

1. L.Banachowski, K.Diks, W.Rytter, Algorytmy i struktury danych, PWN, 2018

C++ Programming Language

Course description

Basic Information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of studies : full-time studies

Study profile : general academic

Required : optional

Education cycle : 2022/23

Course code : UJ.WMIIANS.140.03342.22

Languages : Polish

Disciplines : Computer Science

ISCED Classification : 0613 Software and applications development and analysis

USOS Code : WMI.TCS.JPC+.S

Course coordinator

Krzysztof Turowski

Course instructor

Krzysztof Turowski

Period Semester 3	Form of verification of learning outcomes graded credit Course format and hours lecture: 15 laboratory classes: 15	ECTS credits 3.0

Learning outcomes for the course

Code	Outcomes in the area of:	Directional learning outcomes	Verification methods

Code	Outcomes in the area of:	Directional learning outcomes	Verification methods
Knowledge – Student knows and understands:			
W1	- purpose of syntactic constructs of C++ language (in the C++14 standard) - object-oriented and generic programming techniques - basic functions of the standard library	IAN_K1_W04, IAN_K1_W05	graded credit
Skills – Student can:			
U1	- use appropriate syntactic constructs for the problem when writing code in C++ - find programming errors and optimize programs written in C++ - predict the behavior of programs written in C++ - use the standard library	IAN_K1_U03, IAN_K1_U05, IAN_K1_U08	graded credit

ECTS credit balance

Student's activity form	Average number of hours* allocated to completed activity types	
lecture	15	
laboratory classes	15	
preparation for exam	30	
preparation for classes	30	
Total student workload	Hours 90	ECTS 3.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	1. Program compilation stages 2. Name lookup and visibility scopes 3. Types, references and object lifetime 4. Conversions - explicit, implicit and programmer-defined 5. Compile-time constant expressions 6. Metaprogramming and templates 7. Function overloading 8. Expression initialization 9. Expression evaluation 10. Standard library architecture	W1, U1

Extended information

Teaching methods:

multimedia lecture, discussion, case studies, problem solving, laboratory classes, consultations

Class type	Assessment forms	Course completion conditions
lecture	graded credit	Positive grade from the final test, preceded by admission based on attendance in classes
laboratory classes	graded credit	Positive grade from the final test, preceded by admission based on attendance in classes

Prerequisites and additional requirements

Programming Basics: - basic elements and syntax of C language - main functions from the standard C library - number representation in computer memory - simple algorithms using basic data structures - basic concepts of computational complexity

Literature

Required

1. Bjarne Stroustrup - A Tour of C++ (Second edition), Addison-Wesley, 2018

Foundation of Analytic Information Theory, Machine Learning, and AI

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Level of education : first-cycle studies

Form of studies : full-time studies

Study profile : general academic

Mandatory status : optional

Education cycle : 2022/23

Course code : N/A

Languages of instruction : English

Disciplines : Computer Science

ISCED classification : N/A

USOS code : WMI.TCS.FAITML.S

Course coordinator

Wojciech Szpankowski

Course instructor

Wojciech Szpankowski

Period Semester 5	Form of verification of learning outcomes graded credit Form of teaching and hours lecture with elements of a discussion class: 60	Number of ECTS credits 6.0

Extended information

This course will be run as a research seminar with several talks in the first two weeks followed by students presenting their research. The talks will be based on previous research presentations delivered over the last 5

years in various venues by me and my collaborators. In short, at the beginning I will cover information theory starting with some philosophical discussion on "what is information" and then moving to analytic information theory. Then we discuss structural and temporal information and finally connect analytic information theory and analytic learning theory. In the next week, together with my postdoc we present elements of machine learning and AI. We focus on online learning in the first four lectures with the last on large language models of AI.

At the end of the course (November 4-8) students will be asked to give presentations on their favorite research topics. The grade will be based on participations and the final presentations.

This will be a join seminar with Center for Science of Information (CSoI)

The talks will be given in English and will be recorded.

Approximate Plan of Talks:

1. Organizational Meeting
2. What is Information?
3. Analytic Information Theory I & II
4. Structural and Temporal Information
5. From Analytic Information Theory to Analytic Learning Theory.
6. Learning from Expert Advice
7. Minimax Regret in Online Learning
8. Special Losses
9. Computational Efficiency, Hybrid Settings, and Randomization
10. Introduction to Large Language Models

Data Engineering

Course description

Basic Information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of studies : full-time studies

Study profile : general academic

Required : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.120.03338.22

Languages : Polish

Disciplines : Computer Science

ISCED Classification : 0612 Database and network design and administration

USOS Code : WMI.TCS.ID.OL

Course coordinator

Katarzyna Grygiel

Course instructor

Katarzyna Grygiel

Period Semester 2	Form of verification of learning outcomes exam Course format and hours lecture: 30 laboratory classes: 30	ECTS credits 6.0

Learning outcomes for the course

Code	Outcomes in the area of:	Directional learning outcomes	Verification methods

Code	Outcomes in the area of:	Directional learning outcomes	Verification methods
Knowledge – Student knows and understands:			
W1	- relational data model - structured query language SQL - architecture of modern relational database systems - characteristics, purpose and mechanisms of concurrent transaction execution - mechanisms for failure prevention and recovering databases in case of failure - techniques for modeling conceptual schemas (entity model) - normalization methods in the relational data model - purpose and goal of using data warehouses	IAN_K1_W03, IAN_K1_W07, IAN_K1_W08, IAN_K1_W14	written / oral exam
Skills – Student can:			
U1	- use SQL language and related procedural languages - effectively design and implement information systems using databases - secure and maintain database systems	IAN_K1_U03, IAN_K1_U04, IAN_K1_U11, IAN_K1_U12, IAN_K1_U13, IAN_K1_U14, IAN_K1_U16, IAN_K1_U17, IAN_K1_U18, IAN_K1_U19, IAN_K1_U21	project, credit, written / oral exam
Social competences – Student is ready to:			
K1	- cooperate on modeling, implementing and maintaining information systems - critically analyze the design and applied security measures of information systems	IAN_K1_K01, IAN_K1_K02	project, credit, written / oral exam

ECTS credit balance

Student's activity form	Average number of hours* allocated to completed activity types

lecture	30
laboratory classes	30
independently solving computer problems	60
project preparation	60

Total student workload	Hours 180	ECTS 6.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	1. Characteristics of relational databases. 2. Modeling conceptual schemas and implementation schemas in the relational model. 3. Entity relationship model and its transformation to the relational model. 4. Data organization in modern database systems. 5. Characteristics, purpose and techniques of concurrent transaction execution. 6. Database recovery methods after failure. 7. Normalization of relational databases. 8. Structured Query Language SQL. 9. Query optimization. 10. Data warehouses, Big Data and noSQL.	W1, U1, K1

Extended information

Teaching methods:

project method, multimedia lecture, discussion, case studies, problem solving, laboratory classes, consultations

Class type	Assessment forms	Course completion conditions
lecture	written / oral exam	Positive grade from the exam preceded by admission based on a positive grade from exercises/laboratory
laboratory classes	project, credit	Solving independent programming tasks as well as implementation of a team project.

Prerequisites and additional requirements

Formal Methods in Computer Science: - set theory with particular emphasis on the concept of relations Programming Basics: - simple algorithms using basic data structures - basic programming skills in C++ language - basic concepts of computational complexity

Literature

Required

1. J.D. Ullman, J. Widom, Podstawowy wykład z systemów baz danych, WNT, W-wa, 2000 (seria: Klasyka Informatyki)

Additional

1. J. Celko, SQL zaawansowane techniki programowania, Mikom, 1999
2. M. Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, O'Reilly Media, 2017

Software Engineering

Course Description

Basic Information

Field of Study : Analytical Computer Science

Path : -

Organizational Unit : Faculty of Mathematics and Computer Science

Education Level : first-cycle studies

Form of Studies : full-time studies

Study Profile : general academic

Obligatory Status : mandatory

Education Cycle : 2022/23

Course Code : UJ.WMIIANS.180.01923.22

Languages of Instruction : Polish

Disciplines : Computer Science

ISCED Classification : 0613 Software and applications development and analysis

USOS Code : WMI.TCS.IO.OL

Course Coordinator

Bartosz Walczak

Course Instructor

Bartosz Walczak

Period Semester 4	Form of verification of learning outcomes graded credit Form of instruction and hours laboratory classes: 30	Number of ECTS points 3.0

Learning Outcomes for the Course

Code	Effects in terms of	Field-specific learning outcomes	Verification methods

Code	Effects in terms of	Field-specific learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	the software development process and tools and environments for its design, testing, versioning, and maintenance	IAN_K1_W03, IAN_K1_W15	graded credit
Skills – The student can:			
U1	prepare, design, and implement a simple IT system using appropriate methods and tools, both independently and in a team	IAN_K1_U04, IAN_K1_U11, IAN_K1_U18, IAN_K1_U20	graded credit
U2	design software according to the object-oriented methodology, using object modeling tools and design patterns	IAN_K1_U15, IAN_K1_U16, IAN_K1_U21	graded credit
Social competences – The student is ready to:			
K1	work in a team, ask questions, engage in discussions, and critically evaluate statements and opinions	IAN_K1_K01, IAN_K1_K02	graded credit

ECTS Credit Balance

Student activity form	Average number of hours* dedicated to completed activity types	
laboratory classes	30	
project preparation	40	
test preparation	10	
problem solving	5	
Total student workload	Number of hours 85	ECTS 3.0

* hour (lesson) means 45 minutes

Program Content

No.	Program Content	Learning outcomes for the course
1.	1) Software development process 2) Requirements analysis, object modeling, and UML language 3) Principles and patterns of object-oriented software design 4) Software architecture design 5) Software testing 6) Software refactoring and TDD 7) Software versioning and continuous integration	W1, U1, U2, K1

Extended Information

Teaching Methods:

laboratory classes

Type of classes	Forms of credit	Course completion conditions
laboratory classes	graded credit	appropriate activity during classes or in project preparation, solving an adequate number of tasks, sufficiently high result on the test or project defense

Prerequisites and Additional Requirements

1. Ability to program in Java
2. Ability to work in a programming environment
3. Knowledge of basic concepts of object-oriented programming

Literature

Required

1. Original course – material covered during classes

Additional

1. R. C. Martin: "Zwinne wytwarzanie oprogramowania. Najlepsze zasady, wzorce i praktyki"

Algebraic Methods in Computer Science

Course Description

Basic Information

Field of Study : Analytical Computer Science

Path : -

Organizational Unit : Faculty of Mathematics and Computer Science

Education Level : first-cycle studies

Form of Studies : full-time studies

Study Profile : general academic

Obligatory Status : mandatory

Education Cycle : 2022/23

Course Code : UJ.WMIIANS.110.03334.22

Languages of Instruction : Polish

Disciplines : Computer Science, Mathematics

ISCED Classification : 0541 Mathematics, 0613 Software and applications development and analysis

USOS Code : WMI.TCS.MAI.OL

Course Coordinator

Paweł Idziak

Course Instructors

Paweł Idziak, Andrzej Pezarski

Period Semester 1	Form of verification of learning outcomes exam Form of instruction and hours lecture: 45 tutorials: 60	Number of ECTS credits 8.0

Learning Outcomes for the Course

Code	Effects in terms of	Field-specific learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	basic algebraic, geometric, and number theory concepts and their applications in computer science.	IAN_K1_W01	written exam, credit
W2	basic algorithms of algebra and number theory.	IAN_K1_W09, IAN_K1_W12	written exam, credit
Skills – The student can:			
U1	present mathematical reasoning in a comprehensible way, formulate definitions and theorems, and apply them in computer science practice.	IAN_K1_U01, IAN_K1_U02	written exam, credit
U2	apply mathematical knowledge to model simple tasks related to computer science	IAN_K1_U01	written exam, credit
Social competences – The student is ready to:			
K1	treat with reservation opinions and statements that have not been sufficiently and correctly justified; can precisely formulate questions to analyze a given topic.	IAN_K1_K01, IAN_K1_K05	written exam, credit
K2	critically evaluate their knowledge.	IAN_K1_K01, IAN_K1_K05	written exam, credit

ECTS Credit Balance

Student activity form	Average number of hours* dedicated to completed activity types
lecture	45
tutorials	60
preparation for tutorials	90
exam preparation	43
participation in exam	2

Total student workload	Number of hours 240	ECTS 8.0

* hour (lesson) means 45 minutes

Program Content

No.	Program Content	Learning outcomes for the course
1.	Permutations and groups.	W1, W2, U1, U2, K1, K2
2.	Fields, complex numbers.	W1, W2, U1, U2, K1, K2
3.	Numerical matrices; Determinants, inverse matrix; Vector and matrix norms.	W1, W2, U1, U2, K1, K2
4.	Linear spaces; Linear transformations; Linear functionals.	W1, W2, U1, U2, K1, K2
5.	Systems of linear equations; Image, rank, and kernel of a matrix.	W1, W2, U1, U2, K1, K2
6.	Eigenvalue problems of a linear operator (matrix); Diagonalization.	W1, W2, U1, U2, K1, K2
7.	Euclidean and unitary spaces.	W1, W2, U1, U2, K1, K2
8.	Finite fields, RSA and discrete logarithm	W1, W2, U1, U2, K1, K2
9.	Bilinear and quadratic forms.	W1, W2, U1, U2, K1, K2

Extended Information

Teaching Methods:

conventional lecture, multimedia presentation lecture, subject tutorials

Type of classes	Forms of credit	Course completion conditions
lecture	written exam	obtaining more than 50% of points in the weighted average of the written exam (with weight 40%) and tutorial credit (with weight 60%)
tutorials	credit	activity in class, including solving homework assignments; passing written tests

Prerequisites and Additional Requirements

The course "Formal Methods in Computer Science" must be completed simultaneously or earlier

Literature

Required

1. Herdegen A., "Wykłady z algebry liniowej i geometrii", Discepto, Kraków, 2005

2. Kostrikin A., "Zbiór zadań z algebry", PWN, Warszawa, 1995

Additional

1. Kostrikin A., "Wstęp do algebry 1: Podstawy algebry", PWN, Warszawa, 2004
2. Kostrikin A., "Wstęp do algebry 2: Algebra liniowa", PWN, Warszawa, 2004
3. Kostrikin A., "Wstęp do algebry 3: Podstawowe struktury algebraiczne", PWN, Warszawa, 2005

Discrete Mathematics

Course Description

Basic Information

Field of Study : Analytical Computer Science

Path : -

Organizational Unit : Faculty of Mathematics and Computer Science

Education Level : first-cycle studies

Form of Studies : full-time studies

Study Profile : general academic

Obligatory Status : mandatory

Education Cycle : 2022/23

Course Code : UJ.WMIIANS.120.01914.22

Languages of Instruction : Polish

Course Related to Scientific Research : Yes

Disciplines : Computer Science, Mathematics

ISCED Classification : 0541 Mathematics, 0613 Software and applications development and analysis

USOS Code : WMI.TCS.MD.OL

Course Coordinator

Tomasz Krawczyk

Course Instructor

Tomasz Krawczyk

Period Semester 2	Form of verification of learning outcomes exam Form of instruction and hours lecture: 45 tutorials: 45	Number of ECTS credits 8.0

Learning Outcomes for the Course

Code	Effects in terms of	Field-specific learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	the most important concepts and theorems in combinatorics and graph theory, especially those listed in the Course content field.	IAN_K1_W02	graded credit, written/oral exam
Skills – The student can:			
U1	- define basic concepts of discrete mathematics and illustrate them with simple examples - formulate the most important theorems of discrete mathematics and illustrate them with simple examples - present mathematical reasoning in an understandable way - use combinatorial structures in formulating and solving computer science problems - solve a simple combinatorial problem and present the solution orally and in writing - present the topics discussed in class and formulate questions to better understand the subject	IAN_K1_U01, IAN_K1_U02, IAN_K1_U21, IAN_K1_U22	graded credit, written/oral exam
Social competences – The student is ready to:			
K1	approach with appropriate reservation opinions and statements that have not been sufficiently justified.	IAN_K1_K01	graded credit

ECTS Credit Balance

Student activity form	Average number of hours* dedicated to completed activity types
lecture	45
tutorials	45
preparation for tutorials	105

exam preparation	43
participation in exam	2

Total student workload	Number of hours 240	ECTS 8.0

* hour (lesson) means 45 minutes

Course content

No.	Program Content	Learning outcomes for the course
1.	1. Induction, recursion. 2. Counting: binomial coefficients, Stirling numbers, Bell numbers, Catalan numbers, and others. 3. Generating functions. Solving recurrence relations. 4. Partial orders. Dilworth's theorem. 5. Sperner families, Erdos-Ko-Rado theorem. 6. Ramsey's theorem. 7. Flow networks. 8. Graph theory: * trees, cycles, * bipartite graphs, matchings, * k-connectivity, Menger's theorem, * graph coloring, Brooks' theorem, * planar graphs, geometric intersection graphs, * relationships between coloring number, chromatic number, list chromatic number, and other graph parameters.	W1, U1, K1

Extended Information

Teaching Methods:

conventional lecture, problem solving, subject tutorials

Type of classes	Forms of credit	Course completion conditions
lecture	written/oral exam	positive grade on the exam, preceded by admission to it based on a positive grade from tutorials
tutorials	graded credit	activity in class, solving homework assignments

Prerequisites and Additional Requirements

completed courses in Formal Methods in Computer Science and Algebraic Methods in Computer Science

Literature

Required

1. V.Bryant, "Aspekty kombinatoryki", Wydawnictwa Naukowo-Techniczne 1977.

2. R.L.Graham, D.E.Knuth, O.Patashnik, "Matematyka Konkretna", Państwowe Wydawnictwo Naukowe, Warszawa 1996.
3. W.Lipski, "Kombinatoryka dla programistów", Wydawnictwa Naukowo-Techniczne 2004.
4. K.A.Ross, Ch.R.B.Wright, "Matematyka Dyskretna", Państwowe Wydawnictwo Naukowe, Warszawa 1996.
5. Z.Palka, A.Ruciński, "Wykłady z kombinatoryki", Wydawnictwa Naukowo-Techniczne, Warszawa 1998.
6. R.J.Wilson, "Wprowadzenie do teorii grafów", Państwowe Wydawnictwo Naukowe, Warszawa 1985.

Formal Methods in Computer Science

Course Description

Basic Information

Field of Study : Analytical Computer Science

Path : -

Organizational Unit : Faculty of Mathematics and Computer Science

Education Level : first-cycle studies

Form of Studies : full-time studies

Study Profile : general academic

Obligatory Status : mandatory

Education Cycle : 2022/23

Course Code : UJ.WMIIANS.110.03335.22

Languages of Instruction : Polish

Course Related to Scientific Research : Yes

Disciplines : Computer Science, Mathematics

ISCED Classification : 0541 Mathematics, 0613 Software and applications development and analysis

USOS Code : WMI.TCS.MFI.OL

Course Coordinator

Marek Zaionc

Course Instructor

Marek Zaionc

Period Semester 1	Form of verification of learning outcomes exam Form of instruction and hours lecture: 60 tutorials: 60	Number of ECTS credits 10.0

Learning Outcomes for the Course

Code	Effects in terms of	Field-specific learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	formal methods in computer science; discrete and probabilistic methods for modeling computer science issues	IAN_K1_W02	written exam, written credit
Skills – The student can:			
U1	apply mathematical knowledge to model simple tasks related to computer science	IAN_K1_U02	written exam, written credit
U2	present correct mathematical reasoning in an understandable way, formulate definitions and theorems	IAN_K1_U01, IAN_K1_U02	written exam, written credit
Social competences – The student is ready to:			
K1	approach with appropriate reservation opinions and statements that have not been sufficiently and correctly justified; precisely formulate questions for the analysis of a given topic	IAN_K1_K01	written exam, written credit

ECTS Credit Balance

Student activity form	Average number of hours* dedicated to completed activity types
lecture	60
tutorials	60
preparation for tutorials	55
exam preparation	28
participation in exam	2

problem solving	55	
Total student workload	Number of hours 260	ECTS 10.0

* hour (lesson) means 45 minutes

Program Content

No.	Program Content	Learning outcomes for the course
1.	Set theory axioms, sum and pair axioms. Cartesian product, relations, equivalence relations, set partitions. Von Neumann construction of natural numbers, induction theorem, definition by induction, well-ordering principle, and the construction of integers, rational and real numbers. Basic theorems of cardinality theory. Theory of ordered sets, linearly ordered sets, well-ordered sets, basic theorems in this area.	W1, U1, U2, K1

Extended Information

Teaching Methods:

conventional lecture, multimedia presentation lecture, problem solving, subject tutorials

Type of classes	Forms of credit	Course completion conditions
lecture	written exam	The exam will be in the form of a test. Admission to the exam test requires obtaining a positive credit. The final grade consists in 50% of points from the previously obtained credit and in 50% of points from the exam test. GRADES FROM THE RESIT EXAM: The resit exam will be in the form of a test. All persons who did not pass the exam and also persons who did not get credit are admitted to the resit exam. The final grade after the resit exam consists in 40% of points previously earned for credit and in 60% of points from the resit exam test. For persons who have not previously earned credit and who want to take the resit test, the final grade after the resit exam also becomes the credit grade.
tutorials	written credit	The credit grade consists of grades from tests 2 x 40p plus 20p for activity in tutorials. Grading scale: from 0 to 50 unsatisfactory; from 51 to 60 satisfactory; from 61 to 70 satisfactory+; from 71 to 80 good; from 81 to 90 good+; from 91 to 100 very good.

Prerequisites and Additional Requirements

NONE

Literature

Required

1. H.Rasiowa, "Wstep do matematyki wspólczesnej", PWN, Warszawa 1971, 1984, 1998
2. K. Kuratowski, A. Mostowski, "Teoria mnogości", PWN, Warszawa, 1978

Models of Computation

Course Description

Basic Information

Field of Study : Analytical Computer Science

Path : -

Organizational Unit : Faculty of Mathematics and Computer Science

Education Level : first-cycle studies

Form of Studies : full-time studies

Study Profile : general academic

Obligatory Status : mandatory

Education Cycle : 2022/23

Course Code : UJ.WMIIANS.180.03346.22

Languages of Instruction : Polish

Course Related to Scientific Research : Yes

Disciplines : Computer Science

ISCED Classification : 0613 Software and applications development and analysis

USOS Code : WMI.TCS.MO.OL

Course Coordinator

Michał Wrona

Course Instructors

Michał Wrona, Maciej Ślusarek

Period Semester 4	Form of verification of learning outcomes exam Form of instruction and hours lecture: 30 tutorials: 30	Number of ECTS credits 6.0

Learning Outcomes for the Course

Code	Effects in terms of	Field-specific learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	what a formal language is and the basic facts about formal languages,	IAN_K1_W02	written exam, graded credit
W2	the basic tools: minimization of finite automata, mutual simulations of equivalent models, pumping lemmas, diagonal method	IAN_K1_W11	written exam, graded credit
W3	the concept of undecidability and basic computational complexity classes	IAN_K1_W02, IAN_K1_W11	written exam, graded credit
Skills – The student can:			
U1	define a model describing a formal language and place the class of languages described by the defined model in the language hierarchy	IAN_K1_U01, IAN_K1_U04, IAN_K1_U11, IAN_K1_U17, IAN_K1_U21	written exam, graded credit
U2	select an appropriate model for the problem being solved	IAN_K1_U01, IAN_K1_U04, IAN_K1_U11, IAN_K1_U17, IAN_K1_U21	written exam, graded credit
U3	construct finite automata, context-free grammars, and Turing machines	IAN_K1_U09	written exam, graded credit
Social competences – The student is ready to:			

Code	Effects in terms of	Field-specific learning outcomes	Verification methods
K1	prove theorems in computability theory; understand the profound implications of computability theory for broadly defined science and philosophy, e.g., knows and understands Church's thesis	IAN_K1_K01	written exam, graded credit

ECTS Credit Balance

Student activity form	Average number of hours* dedicated to completed activity types	
lecture	30	
tutorials	30	
preparation for tutorials	90	
exam preparation	28	
participation in exam	2	
Total student workload	Number of hours 180	ECTS credits 6.0

* hour (lesson) means 45 minutes

Program Content

No.	Program Content	Learning outcomes for the course
1.	1. Formal languages and their properties. 2. Finite automata and regular expressions. 3. Pumping lemma and Myhill-Nerode theorem. 4. Minimization of finite automata. 5. Properties of regular languages; problems and algorithms. 6. Grammars and context-free languages; stack automata. 7. Pumping lemma for context-free languages and properties of context-free languages. 8. Deterministic stack automata. 9. Turing machines; recursive and recursively enumerable languages. 10. Universal Turing machine; halting problem and undecidable problems, Rice's theorem. 11. Basics of computational complexity: P, NP, coNP, PSPACE.	W1, W2, W3, U1, U2, U3, K1

Extended Information

Teaching Methods:

conventional lecture, problem solving, subject tutorials

Type of classes	Forms of credit	Course completion conditions
lecture	written exam	positive grade on the exam, preceded by admission to it based on a positive grade from tutorials
tutorials	graded credit	solving problems at the board, two tests

Prerequisites and Additional Requirements

Formal Methods in Computer Science

Literature

Required

1. J. Hopcroft, J. Ullman, "Introduction to Automata Theory, Languages, and Computation" (1st ed.), Addison-Wesley, 1979

Programming Methods

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of study : full-time studies

Study profile : general academic

Mandatory status : obligatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.120.03269.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.MP.OL

Course coordinator

Maciej Ślusarek

Course instructors

Maciej Ślusarek, Iwona Cieślik

Period Semester 2	Form of verification of learning outcomes exam Form of instruction and hours lecture: 30 laboratory exercises: 30	Number of ECTS credits 6.0

Learning outcomes for the course

Code	Outcomes in the field of	Field-specific learning outcomes	Verification methods

Code	Outcomes in the field of	Field-specific learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	basic data structures (trees, graphs, hash tables) and methods of their implementation	IAN_K1_W04, IAN_K1_W06	written exam, credit
W2	selected techniques of algorithm construction	IAN_K1_W06	written exam, credit
W3	basic techniques of sorting and searching data	IAN_K1_W06	written exam, credit
Skills – The student can:			
U1	use basic data structures to describe simple problems presented in natural language	IAN_K1_U01, IAN_K1_U07, IAN_K1_U08	written exam, credit
U2	design and implement algorithms using basic data structures and selected programming techniques	IAN_K1_U03, IAN_K1_U05, IAN_K1_U06, IAN_K1_U07, IAN_K1_U08	written exam, credit
U3	write a program in a readable way, test it at a basic level, look for errors and optimize it	IAN_K1_U17	written exam, credit
U4	propose a solution for a simple algorithmic problem by choosing the appropriate method for its solution	IAN_K1_U11, IAN_K1_U21	written exam, credit
U5	orally and in writing present the development of a solution to a simple problem	IAN_K1_U21	written exam, credit
Social competences – The student is ready to:			
K1	precisely formulate questions that serve to deepen or supplement their own understanding of a given topic	IAN_K1_K01	written exam, credit

ECTS credits balance

Student activity form	Average number of hours* devoted to completed activity types
lecture	30
laboratory exercises	30
independent solving of computer tasks	60
preparation for classes	30
preparation for exam	30

Total student workload	Number of hours 180	ECTS credits 6.0

* hour (lesson) means 45 minutes

Program content

No.	Program content	Learning outcomes for the course
1.	1. Computational complexity of algorithms – definition, notation, comparison of complexity functions. The concept of an abstract data type (ADT). Basic data structures: list, stack, queue, priority queue, dictionary; array, pointer, cursor implementation. Examples: binary search, topological sorting, memory management. 2. Tree structures: trees, binary trees, representations, elementary and advanced traversal algorithms, binary search tree. 3. Amortized complexity, dynamic arrays, hashing (basics). 4. Graphs: representation, BFS and DFS traversal, connected components, cycles, topological sorting with DFS, Euler cycle. 5. Recursion, conversion to iteration, examples: DFS with stack, backtracking. 6. Divide and conquer method, fast multiplication, merge sort, master theorem (simplified version). 7. Quicksort, variants (Hoare, Lomuto), non-recursive version. 8. Binary heap, heapsort, order statistics, counting tree. 9. Radix sort, lower bound on sorting complexity. 10. Dynamic programming – introduction. 11. Greedy algorithms – introduction.	W1, W2, W3, U1, U2, U3, U4, U5, K1

Extended information

Teaching methods:

conventional lecture, laboratory exercises

Type of classes	Credit forms	Course credit conditions

Type of classes	Credit forms	Course credit conditions
lecture	written exam	Positive grade from the exam. Admission to the exam under the condition of a positive grade from the laboratory. The final grade is the average of the laboratory grade and the exam.
laboratory exercises	credit	Laboratory credit based on credit programs, homework assignments, and tests.

Prerequisites and additional requirements

Programming Basics

Literature

Required

1. T.H. Cormen, Ch.E. Leiserson, R.L. Rivest, C. Stein, Wprowadzenie do algorytmów, wydanie III, PWN, 2012

Additional

1. L.Banachowski, K.Diks, W.Rytter, Algorytmy i struktury danych, PWN, 2018
2. D. Knuth, Sztuka programowania, tom 1 i 3, WNT, 2002
3. A.V.Aho, J.E.Hopcroft, J.D.Ullman, Projektowanie i analiza algorytmów, PWN 1985, Helion 2003.

Probabilistic Methods in Computer Science

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of study : full-time studies

Study profile : general academic

Mandatory status : obligatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.140.03339.22

Languages of instruction : Polish

Disciplines : Computer Science, Mathematics

ISCED classification : 0541 Mathematics, 0613 Software and applications development and analysis

USOS code : WMI.TCS.MPI.OL

Course coordinator

Piotr Micek

Course instructors

Piotr Micek

Period Semester 3	Form of verification of learning outcomes exam Form of instruction and hours lecture: 30 tutorials: 30	Number of ECTS credits 6.0

Educational aims for the course

C1	Familiarizing students with concepts and theorems from the field of probability calculus and developing the ability to use them freely

Learning outcomes for the course

Code	Outcomes in the field of	Field-specific learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	classical distributions of random variables and their modifications.	IAN_K1_W01, IAN_K1_W02	oral exam, credit
W2	the principle of linearity of expected value (and can use it in solving problems).	IAN_K1_W01, IAN_K1_W02	oral exam, credit
W3	simple random processes: walks, branching processes, Markov chains.	IAN_K1_W01, IAN_K1_W02	oral exam, credit
Skills – The student can:			
U1	model a probability space for described random experiments.	IAN_K1_U01, IAN_K1_U02	oral exam, credit
U2	apply simulation of random variables in computer science.	IAN_K1_U01, IAN_K1_U02	oral exam, credit

ECTS credits balance

Student activity form	Average number of hours* devoted to completed activity types	
lecture	30	
tutorials	30	
preparation for tutorials	90	
preparation for exam	29	
participation in exam	1	
Total student workload	Number of hours 180	ECTS credits 6.0

* hour (lesson) means 45 minutes

Program content

No.	Program content	Learning outcomes for the course
1.	1. Axioms of probability calculus. 2. Conditional probability and independence of events. 3. Random variable: its distribution and cumulative distribution function. 4. Discrete random variables and their parameters. 5. Random walks. 6. Continuous random variables and their parameters. 7. Buffon's needle and geometric probability. 8. Generating functions of random variables. 9. Generating functions for random walks and branching processes. 10. Limit theorems and characteristic function. 11. Poisson process and Markov chain. 12. Theory of codes and entropy. 13. Simulation of random variables. Elements of statistics.	W1, W2, W3, U1, U2

Extended information

Teaching methods:

conventional lecture, multimedia presentation lecture, subject exercises

Type of classes	Credit forms	Course credit conditions
lecture	oral exam	Obtaining an appropriate number of points for the exam and tutorials
tutorials	credit	Obtaining an appropriate number of points for exercises

Prerequisites and additional requirements

knowledge of discrete mathematics and mathematical analysis

Literature

Required

1. 1. Geoffrey G. Grimmet, David R. Stirzaker, Probability and Random Processes, Oxford University Press 2001
   2. Jacek Jakubowski, Rafał Sztencel, Wstęp do teorii prawdopodobieństwa, Wydawnictwo SCRIPT 2004
   3. Agnieszka i Edmund Plucińscy, Probabilistyka, Wydawnictwo Naukowo-Techniczne Warszawa
   4. Sheldon Ross, A first course in probability, 8th edition

Combinatorial Optimization

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of study : full-time studies

Study profile : general academic

Mandatory status : elective

Education cycle : 2022/23

Course code : UJ.WMIIANS.1300.03364.22

Languages of instruction : Polish

Course related to scientific research : Yes

Disciplines : Computer Science, Mathematics

ISCED classification : 0588 Interdisciplinary programs and qualifications involving natural sciences, mathematics and statistics

USOS code : WMI.TCS.OK.S

Course coordinator

Bartłomiej Bosek

Course instructors

Bartłomiej Bosek

Periods Semester 5, Semester 6	Form of verification of learning outcomes grade credit Form of instruction and hours seminar: 30	Number of ECTS credits 3.0

Learning outcomes for the course

Code	Outcomes in the field of	Field-specific learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	formal methods of computer science; discrete and probabilistic methods modeling computer science issues	IAN_K1_W02	presentation
Skills – The student can:			
U1	acquire and clearly present knowledge from professional literature	IAN_K1_U01, IAN_K1_U02, IAN_K1_U17, IAN_K1_U21, IAN_K1_U23, IAN_K1_U24, IAN_K1_U26	presentation
Social competences – The student is ready to:			
K1	know the limitations of their own knowledge and understand the need for further education	IAN_K1_K01, IAN_K1_K02	presentation

ECTS credits balance

Student activity form	Average number of hours* devoted to completed activity types	
seminar	30	
preparation of presentation	60	
Total student workload	Number of hours 90	ECTS credits 3.0

* hour (lesson) means 45 minutes

Program content

No.	Program content	Learning outcomes for the course
1.	This is a seminar whose subject matter concerns combinatorial optimization. In particular, we are interested in the following topics: 1) Matchings in graphs. 2) Packing objects on a plane. 3) Partial orders, dimension, width, partitions. 4) Coloring of graphs and partial orders.	W1, U1, K1

Extended information

Teaching methods:

seminar

Type of classes	Credit forms	Course credit conditions
seminar	presentation	positive evaluation of presentation

Prerequisites and additional requirements

The student should have mastered the basics of mathematics, combinatorics, and algorithms. They should know the concept of mathematical proof and be proficient in using formal mathematical notation. Passive knowledge of English at a level sufficient for independent reading of scientific texts.

Literature

Required

1. Articles from leading international journals and conferences.

Protection of Intellectual Property

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of study : full-time studies

Study profile : general academic

Mandatory status : obligatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.140.00005.22

Languages of instruction : Polish

Course related to scientific research : Yes

Disciplines : Legal Sciences

ISCED classification : 0421 Law

USOS code : WMI.TCS.OWI.OM, WMI.TCS.OWI.OL

Course coordinator

Ewa Laskowska-Litak

Course instructors

Nicholas Ghazal, Ewa Laskowska-Litak

Period Semester 3	Form of verification of learning outcomes credit Form of instruction and hours lecture: 5	Number of ECTS credits 1.0

Educational aims for the course

C1	Familiarizing the student with basic concepts in the field of intellectual property law.

Learning outcomes for the course

Code	Outcomes in the field of	Field-specific learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	principles of intellectual property protection.	IAN_K1_W17	credit
W2	principles of trading in intangible goods.	IAN_K1_W17	credit
Skills – The student can:			
U1	assess whether a given way of using an intangible good is legal.	IAN_K1_U24	credit
U2	use the right of quotation.	IAN_K1_U24	credit
Social competences – The student is ready to:			
K1	conduct business, professional, social activities based on the use of intellectual property goods.	IAN_K1_K05, IAN_K1_K07	credit
K2	conduct activities related to the popularization of intellectual property protection.	IAN_K1_K05, IAN_K1_K06, IAN_K1_K07	credit

ECTS credits balance

Student activity form	Average number of hours* devoted to completed activity types	
lecture	5	
preparation for classes	25	
Total student workload	Number of hours 30	ECTS credits 1.0

* hour (lesson) means 45 minutes

Program content

No.	Program content	Learning outcomes for the course
1.	Introduction to intellectual property law.	W1, W2, U1, K1, K2
2.	Introduction to copyright law issues. Work as a subject of copyright.	W1, W2, U1, U2, K1, K2
3.	Introduction to industrial property law issues with particular emphasis on patent law and trademark law.	W1, W2, U1, K1, K2
4.	Principles of legal use of intangible goods. Freedom of expression and intellectual property rights.	W1, W2, U1, U2, K1, K2
5.	Plagiarism as a manifestation of infringement of the right to authorship of a work.	W1, W2, U1, U2, K1, K2
6.	Appropriation of someone else's scientific findings as a manifestation of violation of personal rights under common law.	W1, W2, U1, K1, K2

Extended information

Teaching methods:

conventional lecture, multimedia presentation lecture, discussion

Type of classes	Credit forms	Course credit conditions
lecture	credit	Participation in the lecture

Prerequisites and additional requirements

None

Literature

Required

1. J. Sieńczyło-Chlabicz (red.), Prawo własności intelektualnej, Warszawa 2018

Additional

1. R. Markiewicz, Ilustrowane prawo autorskie, Warszawa 2018
2. R. Markiewicz, Zabawy z prawem autorskim, Warszawa 2015

Paradigms of Programming Languages

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Level of education : first-cycle studies

Form of studies : full-time studies

Study profile : general academic

Mandatory status : optional

Education cycle : 2022/23

Course code : UJ.WMIIANS.1300.03365.22

Languages of instruction : Polish

Course related to scientific research : Yes

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.PJP.S

Course coordinator

Grzegorz Herman

Course instructor

Grzegorz Herman

Period Semester 5, Semester 6	Form of verification of learning outcomes graded credit Form of teaching and hours seminar: 30	Number of ECTS credits 3.0

Learning outcomes for the course

Code	Effects in the area of	Major learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	contemporary directions of development and achievements in the field of programming languages	IAN_K1_W05	presentation
Skills – The student can:			
U1	obtain and integrate information from scientific publications in English	IAN_K1_U24	presentation
U2	prepare an oral presentation of scientific results	IAN_K1_U21, IAN_K1_U23	presentation
Social competences – The student is ready to:			
K1	understand the limitations of their knowledge and the need for further education	IAN_K1_K05	presentation

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types	
seminar	30	
preparation of a paper	60	
Total student workload	Number of hours 90	ECTS credits 3.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course

No.	Course content	Learning outcomes for the course
1.	"Paradigms of Programming Languages" is a seminar dedicated to programming language design and implementation. Students with a stronger theoretical inclination are invited to independently tackle the latest scientific results in this field (type systems, program correctness proofs, optimization, parallelization, etc.). For those interested in the practical side of the topic, we propose participation in an IT project related to programming language design and compiler implementation.	W1, U1, U2, K1

Extended information

Teaching methods:

seminar

Type of classes	Forms of credit	Course credit requirements
seminar	presentation	Attendance at the seminar. Understanding an English-language scientific paper and preparing a comprehensible oral presentation of the results described in it.

Prerequisites and additional requirements

1. Good knowledge of at least two programming languages. 2. Knowledge of x86_64 computer architecture. 3. Basic information about the program compilation process. 4. Passive knowledge of English at a level sufficient for independent reading of scientific texts. 5. Efficient use of formal mathematical notation. 6. Basic knowledge of functional programming (in any language) is welcome.

Literature

Required

1. (not applicable)

Low-level Language Programming

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Level of education : first-cycle studies

Form of studies : full-time studies

Study profile : general academic

Mandatory status : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.180.03299.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.PN.OL

Course coordinator

Jakub Kozik

Course instructor

Jakub Kozik, Jan Derbisz

Period Semester 4	Form of verification of learning outcomes exam Form of teaching and hours lecture: 30 laboratory classes: 30	Number of ECTS credits 6.0

Educational goals for the course

C1	When we exhaust algorithmic ways to speed up a computer program, we are left with low-level code optimization. Compilers, virtual machines, and code generators become more powerful every year, but to achieve the highest efficiency, we often need to optimize code manually. In this course, you will learn how to manage memory well, why it's worth using cursor structures, and how to live in harmony with cache memory. We will show you how not to ruin the performance of a multithreaded program with poor synchronization. We will teach you how to squeeze every last bit of performance from modern processors using their vector capabilities and many other useful techniques that will expand your programming toolkit.

Learning outcomes for the course

Code	**Effects in the area of**	**Major learning outcomes**	**Verification methods**
Knowledge – The student knows and understands:			
W1	fundamentals of modern processor architecture.	IAN_K1_W13	written exam, graded credit
Skills – The student can:			
U1	program in assembly language using advanced processor functionalities (including SIMD and atomic instructions).	IAN_K1_U03, IAN_K1_U09	written exam, graded credit, programming tasks
U2	use elements of low-level optimization in high-level language programming (C/C++).	IAN_K1_U03, IAN_K1_U05, IAN_K1_U09	written exam, graded credit, programming tasks

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types
lecture	30
laboratory classes	30
independent solving of computer tasks	75
exam preparation	40

Total student workload	Number of hours 175	ECTS credits 6.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	Basics of x86_64 architecture.	W1
2.	Low-level interfaces of Linux system and C and C++ languages.	W1, U1
3.	Elements of processor microarchitecture (including pipeline processing and cache operation).	W1, U2
4.	SIMD (Single Instruction Multiple Data) instructions.	W1, U1, U2
5.	Binary code and programs modifying code (Self Modifying Code).	U1
6.	Low-level aspects of multithreaded programming.	W1, U2

Extended information

Teaching methods:

multimedia lecture, discussion, problem solving, laboratory classes

Type of classes	Forms of credit	Course credit requirements
lecture	written exam	The condition for passing the course is obtaining a positive grade from classes and passing the exam with more than 50% of points.
laboratory classes	graded credit, programming tasks	The condition for passing the classes is obtaining more than 50% of points for completing programming tasks. Activity during classes can improve the grade but does not affect the fact of passing.

Prerequisites and additional requirements

- Ability to program in C and C++. - Knowledge of operating systems basics.

Literature

Required

1. Technical documentation for x86_64 architecture for AMD or Intel processors, available on the manufacturer's website.

Additional

1. Randall Hyde, The Art of Assembly Language Programming (available on the author's website)
2. Agner Fog, Optimization manuals (available on the author's website)

Object-Oriented Programming

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Level of education : first-cycle studies

Form of studies : full-time studies

Study profile : general academic

Mandatory status : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.120.03337.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.POB.OL

Course coordinator

Marcin Kozik

Course instructor

Marcin Kozik

Period Semester 2	Form of verification of learning outcomes graded credit Form of teaching and hours lecture: 45 laboratory classes: 30	Number of ECTS credits 7.0

Learning outcomes for the course

Code	Effects in the area of	Major learning outcomes	Verification methods

Code	Effects in the area of	Major learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	object-oriented programming paradigms and at least two object-oriented programming languages.	IAN_K1_W05, IAN_K1_W06, IAN_K1_W15	graded credit
Skills – The student can:			
U1	use several object-oriented programming languages and tools for versioning, building, and testing programs in these languages.	IAN_K1_U03, IAN_K1_U04, IAN_K1_U05, IAN_K1_U08, IAN_K1_U15, IAN_K1_U20, IAN_K1_U24, IAN_K1_U26	graded credit
Social competences – The student is ready to:			
K1	work independently and in a team on software development using object-oriented programming languages.	IAN_K1_K02	graded credit

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types	
lecture	45	
laboratory classes	30	
independent solving of computer tasks	60	
project preparation	60	
test preparation	15	
Total student workload	Number of hours 210	ECTS credits 7.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	Object-oriented programming techniques with examples in Java and C++ 1. encapsulation and information hiding, classes and subclasses, inheritance, interfaces, polymorphism, class hierarchies (Java, C++) 2. generic types (Java), templates (C++) 3. containers and iterators (Java, C++) 4. exceptions (Java, C++) 5. reflection (Java) and RTTI (C++) 6. threads (Java) 7. input/output (Java, C++) 8. garbage collection (Java) 9. GUI (Java)	W1, U1, K1

Extended information

Teaching methods:

project method, multimedia lecture, problem solving, e-learning methods

Type of classes	Forms of credit	Course credit requirements
lecture	graded credit	The lecture grade is identical to the laboratory grade
laboratory classes	graded credit	The grade consists of results from tests, evaluation of programming task solutions submitted online by students, and evaluation of a mini-project created as part of the course.

Prerequisites and additional requirements

Completed "Programming Basics" lecture.

Literature

Required

1. Effective modern c++ Scott Meyers
2. Effective Java Joshua Block

Programming Basics

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Level of education : first-cycle studies

Form of studies : full-time studies

Study profile : general academic

Mandatory status : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.110.03024.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.PP.OL

Course coordinator

Iwona Cieślik

Course instructor

Iwona Cieślik

Period Semester 1	Form of verification of learning outcomes exam Form of teaching and hours lecture: 30 laboratory classes: 30	Number of ECTS credits 6.0

Educational goals for the course

C1	Developing basic programming skills in C and C++ languages.
C2	Developing basic skills in creating simple algorithms.

Learning outcomes for the course

Code	Effects in the area of	Major learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	the syntax of C and C++ and basic functions from the standard libraries of these languages	IAN_K1_W04, IAN_K1_W05	programming tasks, test
W2	number representations and properties of computer arithmetic	IAN_K1_W13	written exam
W3	basics of algorithms, basic data structures (arrays, lists, trees), their computer representations and operations performed on them, and basic techniques of algorithm construction and analysis	IAN_K1_W06, IAN_K1_W07, IAN_K1_W08	written exam, programming tasks, test
Skills – The student can:			
U1	program in C and C++ languages	IAN_K1_U03, IAN_K1_U05	programming tasks, test
U2	design and implement simple algorithms using basic data structures such as: arrays, strings, pointers, structures, objects, files, linked lists; use these structures to describe simple problems presented in natural language	IAN_K1_U03, IAN_K1_U05, IAN_K1_U07	written exam, programming tasks, test
U3	use basic programming techniques such as function calls, recursion, backtracking; choose the appropriate method	IAN_K1_U03, IAN_K1_U06	written exam, programming tasks, test
U4	write programs in a readable way and analyze their code to find errors	IAN_K1_U03, IAN_K1_U05	programming tasks, test
Social competences – The student is ready to:			
K1	formulate questions to better understand a given topic	IAN_K1_K01	graded credit

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types
lecture	30
laboratory classes	30
independent solving of computer tasks	90
exam preparation	10
test preparation	5
exam participation	3

Total student workload	Number of hours 168	ECTS credits 6.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	Programming in C language: a. basic and complex data types, b. operators, control instructions c. arrays, strings, pointers, functions d. input/output operations, working with files e. dynamic memory allocation f. complex data types g. dynamic data structures (lists, stacks, queues) h. basic functions from the standard library	W1, W3, U1, U2, U3, U4, K1
2.	Basics of object-oriented programming in C++ a. creating classes, public and private methods b. operator overloading c. streams, input/output operations d. dynamic memory allocation	W1, W3, U1, U2, U3, U4, K1
3.	Basics of algorithms a. Euclidean algorithm b. quadratic sorting algorithms c. binary search d. fast exponentiation e. reverse Polish notation f. positional systems and conversion algorithms g. recursion, backtracking algorithms, game tree h. dynamic data structures (lists, stacks, queues)	K1
4.	Theoretical foundations of programming a. Number representations: positional systems, conversion algorithms, sign-magnitude and two's complement systems, fixed-point and floating-point representation, properties of computer arithmetic. b. Example of a digital machine - von Neumann Machine c. Algorithm correctness, invariants d. Basic concepts of computational complexity	W2, W3, K1

Extended information

Teaching methods:

multimedia lecture, discussion, problem solving, laboratory classes

Type of classes	Forms of credit	Course credit requirements
lecture	written exam	The student receives a final grade for the course based on points awarded during classes and points obtained during the written exam. The condition for receiving a positive final grade is obtaining class credit and accumulating a total of at least 60% of points.
laboratory classes	graded credit, programming tasks, test	The student receives a final grade for classes based on points awarded for systematically submitted programming tasks (mandatory and additional) and points obtained on the test. The condition for receiving class credit is submitting at least 70% of programming tasks, including all mandatory tasks, and accumulating a total of 60% of points.

Literature

Required

1. Lecture materials

Additional

1. B.W.Kerninghan, D.M.Ritchie, "The C Programming Language", Prentice Hall, 1988.
2. C.L.Tondo, S.E.Gimpel, "The C Answer Book: Solutions to the Exercises in 'The C Programming Language'", Prentice Hall, 1988.
3. J.Grębosz, Symfonia C++ Standard, Edition "2000" Publishing, Kraków 2008.
4. C and C++ language documentation.
5. J. Tomasiewicz, Zaprzyjaźnij się z algorytmami. Przewodnik dla początkujących i średnio zaawansowanych, PWN

Programming Project 1

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Level of education : first-cycle studies

Form of studies : full-time studies

Study profile : general academic

Mandatory status : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.1100.03359.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code

Course coordinator

Bartłomiej Bosek

Course instructor

Bartłomiej Bosek

Period Semester 5	Form of verification of learning outcomes graded credit Form of teaching and hours laboratory classes: 30	Number of ECTS credits 3.0

Educational goals for the course

C1	During the course, the student will actively participate in a large programming project.

Learning outcomes for the course

Code	Effects in the area of	Major learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	theoretical and practical issues related to large programming projects.	IAN_K1_W03, IAN_K1_W15	project, presentation
Skills – The student can:			
U1	actively participate in a large programming project.	IAN_K1_U03, IAN_K1_U04, IAN_K1_U11, IAN_K1_U17, IAN_K1_U18, IAN_K1_U20, IAN_K1_U21, IAN_K1_U22, IAN_K1_U24, IAN_K1_U26	project, presentation
Social competences – The student is ready to:			
K1	discuss social aspects related to large programming projects.	IAN_K1_K01, IAN_K1_K02, IAN_K1_K03, IAN_K1_K04	project, presentation

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types	
laboratory classes	30	
project preparation	60	
Total student workload	Number of hours 90	ECTS credits 3.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course

No.	Course content	Learning outcomes for the course
1.	During the course, the student: - will become familiar with selected large programming projects - will learn the principles of contributing to such projects - will actively participate in the development of one of the projects	W1, U1, K1

Extended information

Teaching methods:

project method, discussion, consultations

Type of classes	Forms of credit	Course credit requirements
laboratory classes	project, presentation	The student receives a final grade based on points awarded for active participation in classes, active participation in the project, and systematically submitted reports.

Literature

Required

1. not applicable

Programming Project 2

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Level of education : first-cycle studies

Form of studies : full-time studies

Study profile : general academic

Mandatory status : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.1200.03366.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code

Course coordinator

Andrzej Pezarski

Course instructor

Andrzej Pezarski

Period Semester 6	Form of verification of learning outcomes graded credit Form of teaching and hours laboratory classes: 30	Number of ECTS credits 3.0

Educational goals for the course

C1	During the course, the student will actively participate in a large programming project.

Learning outcomes for the course

Code	Effects in the area of	Major learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	theoretical and practical issues related to large programming projects.	IAN_K1_W03, IAN_K1_W15	project, presentation
Skills – The student can:			
U1	actively participate in a large programming project.	IAN_K1_U03, IAN_K1_U04, IAN_K1_U11, IAN_K1_U17, IAN_K1_U18, IAN_K1_U20, IAN_K1_U21, IAN_K1_U22, IAN_K1_U24, IAN_K1_U26	project, presentation
Social competences – The student is ready to:			
K1	discuss social aspects related to large programming projects.	IAN_K1_K01, IAN_K1_K02, IAN_K1_K03, IAN_K1_K04	project, presentation

ECTS credits balance

Student activity form	Average number of hours* dedicated to completed activity types	
laboratory classes	30	
project preparation	60	
Total student workload	Number of hours 90	ECTS credits 3.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course

No.	Course content	Learning outcomes for the course
1.	During the course, the student: - will become familiar with selected large programming projects - will learn the principles of contributing to such projects - will actively participate in the development of one of the projects	W1, U1, K1

Extended information

Teaching methods:

project method, discussion, consultations

Type of classes	Forms of credit	Course credit requirements
laboratory classes	project, presentation	The student receives a final grade based on points awarded for active participation in classes, active participation in the project, and systematically submitted reports.

Literature

Required

1. not applicable

Empty promises of Thomas Schaefer

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Level of education : first-cycle studies

Form of studies : full-time studies

Study profile : general academic

Mandatory status : optional

Education cycle : 2022/23

Course code : N/A

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : N/A

USOS code : WMI.TCS.PPTS.S

Course coordinator

Marcin Kozik

Course instructor

Marcin Kozik
Demian Banakh

Period Semester 4	Form of verification of learning outcomes Exam Form of teaching and hours lecture: 30 tutorials: 30	Number of ECTS credits 6.0

Concurrent Programming

Course description

Basic Information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of studies : full-time studies

Study profile : general academic

Status : optional

Education cycle : 2022/23

Course code : UJ.WMIIANS.1380.03352.22

Languages of instruction : Polish

Course related to scientific research : Yes

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.PW.S

Course coordinator

Maciej Ślusarek

Course instructors

Maciej Ślusarek, Krzysztof Turowski

Terms Semester 4, Semester 5, Semester 6	Verification method of learning outcomes exam Form of instruction and hours lecture: 30 laboratory classes: 30	Number of ECTS credits 6.0

Learning outcomes for the course

Code	Outcomes in terms of	Directional learning outcomes	Verification methods
Knowledge – Student knows and understands:			
W1	basic concepts, models and techniques of parallel computing	IAN_K1_W04, IAN_K1_W08, IAN_K1_W13	written exam, credit
Skills – Student can:			
U1	design and analyze parallel algorithms for selected problems and parallelism models	IAN_K1_U03, IAN_K1_U05, IAN_K1_U11, IAN_K1_U17, IAN_K1_U21	written exam, credit
U2	program in parallel in GPU environment	IAN_K1_U03, IAN_K1_U05, IAN_K1_U09, IAN_K1_U11	written exam, credit

ECTS credits balance

Form of student activity	Average number of hours* dedicated to completed activity types	
lecture	30	
laboratory classes	30	
project preparation	30	
independent solving of computer tasks	60	
exam preparation	30	
Total student workload	Number of hours 180	ECTS credits 6.0

* hour (lesson) means 45 minutes

Program content

No.	Program content	Learning outcomes for the course

No.	Program content	Learning outcomes for the course
1.	1. Basic concepts of concurrent programming 2. Algorithms in the PRAM model: model properties, complexity parameters, basic techniques: doubling, parallel prefix, Euler path technique for trees 3. Selected algorithms in the PRAM model - transitive closure, shortest paths, BFS, connected components 4. Basics of programming in the CUDA system 5. Multi-threaded algorithms in the CILK system 6. Threads in the POSIX standard 7. OpenMP 8. MPI 9. Selected parallel algorithms (parallel prefix, sorting, graph problems, matrix operations) in various concurrent computing models.	W1, U1, U2

Extended information

Teaching methods:

conventional lecture, laboratory classes

Type of classes	Credit forms	Course credit conditions
lecture	written exam	Positive grade from the exam. Admission to the exam subject to a positive grade from the laboratory. The final grade is a weighted average of the laboratory grade and the exam.
laboratory classes	credit	Laboratory credit based on credit programs and project

Prerequisites and additional requirements

Algorithms and data structures 1

Literature

Required

1. A.Grama, A.Gupta, G.Karypis, V.Kumar, Introduction to Parallel Computing (2'nd ed.), Addison-Wesley, 2003

Additional

1. T.H. Cormen, Ch.E. Leiserson, R.L. Rivest, C. Stein, Wprowadzenie do algorytmów, wydanie III, PWN, 2012
2. http://docs.nvidia.com/cuda/cuda-c-programming-guide

Python Programming Language

Course description

Basic Information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle

Form of studies : full-time studies

Study profile : general academic

Status : optional

Education cycle : 2022/23

Course code : UJ.WMIIANS.140.03344.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.JPP.S

Course coordinator

Krzysztof Turowski

Course instructors

Krzysztof Turowski

Term Semester 3	Verification method of learning outcomes grade credit Form of instruction and hours lecture: 15 laboratory classes: 15	Number of ECTS credits 3.0

Learning outcomes for the course

Code	Outcomes in terms of	Directional learning outcomes	Verification methods

Code	Outcomes in terms of	Directional learning outcomes	Verification methods
Knowledge – Student knows and understands:			
W1	the basic elements and data structures of the Python language (listed in the "Program content" field), which enable creating advanced programs and applications in this language.	IAN_K1_W04, IAN_K1_W05	written credit, project
Skills – Student can:			
U1	implement applications and programs in Python.	IAN_K1_U03, IAN_K1_U05, IAN_K1_U08	written credit, project

ECTS credits balance

Form of student activity	Average number of hours* dedicated to completed activity types	
lecture	15	
laboratory classes	15	
project preparation	15	
independent solving of computer tasks	30	
exam preparation	13	
exam participation	2	
Total student workload	Number of hours 90	ECTS credits 3.0

* hour (lesson) means 45 minutes

Program content

No.	Program content	Learning outcomes for the course

No.	Program content	Learning outcomes for the course
1.	The following elements of the Python language will be discussed in class: - basic language syntax (control instructions), - basic Python data structures (lists, tuples, sets, dictionaries), - classes, metaclasses, attribute lookup in objects and classes, - descriptors, - decorators, - code organization in modules, - errors and exceptions, - generators and list comprehensions, - selected elements of standard libraries, network programming libraries. Additionally, the Django framework for creating web applications based on Python will be discussed in class.	W1, U1

Extended information

Teaching methods:

conventional lecture, multimedia presentation lecture, laboratory classes

Type of classes	Credit forms	Course credit conditions
lecture	written credit	positive grade from the exam, preceded by admission to it based on a positive grade from laboratories
laboratory classes	project	implementing programs using the discussed Python language elements, developing a final project

Prerequisites and additional requirements

completed Object-Oriented Programming course

Literature

Required

1. Python language documentation available at https://docs.python.org/

Computer Networks

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle studies

Form of study : full-time studies

Study profile : general academic

Obligatory status : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.140.01925.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0612 Database and network design and administration

USOS code : WMI.TCS.SK.OL

Course coordinator

Grzegorz Gutowski

Course instructor

Grzegorz Gutowski

Period Semester 3	Form of verification of learning outcomes exam Teaching methods and hours lecture: 30 laboratory classes: 30	Number of ECTS credits 6.0

Educational aims of the course

C1	During the course, students will learn theoretical models and practical solutions used in designing, managing, and operating various types of computer networks, and will learn to apply this knowledge in programming projects.

Learning outcomes for the course

Code	Effects in terms of	Directional learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	theoretical and practical issues related to network architectures, technologies, and applications.	IAN_K1_W03, IAN_K1_W16	written exam, project, presentation
Skills – The student can:			
U1	analyze, design, use, and program network solutions.	IAN_K1_U04, IAN_K1_U11, IAN_K1_U12, IAN_K1_U13, IAN_K1_U17, IAN_K1_U18, IAN_K1_U19, IAN_K1_U21	written exam, project, presentation
Social competences – The student is ready to:			
K1	discuss social aspects related to network technologies.	IAN_K1_K01, IAN_K1_K06	written exam, project, presentation

ECTS credits balance

Form of student activity	Average number of hours* devoted to completed types of activities
lecture	30
laboratory classes	30
preparation for classes	15
project preparation	45
problem solving	45
exam preparation	13
exam participation	2

Total student workload	Number of hours 180	ECTS credits 6.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	During the course, the student will encounter the following topics: - communication methods - basic concepts of signal theory - layered models of computer networks - data link layer issues - problems, technologies, and algorithms related to Ethernet networks - problems, technologies, and algorithms related to WiFi networks - network layer issues - problems, technologies, and algorithms related to the Internet - issues related to packet buffering - transport layer issues - problems, technologies, and algorithms used in TCP protocol - issues related to implementation of network protocols - problems, technologies, and algorithms related to HTTP protocol - network communication security issues - issues related to peer-to-peer networks	W1, U1, K1

Extended information

Teaching methods:

conventional lecture, laboratory classes

Type of classes	Forms of credit	Course credit requirements
lecture	written exam	The student receives a final grade based on the sum of points awarded during exercises (0-60) and points obtained during the written exam (0-40). To pass the course, a positive grade from exercises and a minimum of 60 points in total are required.
laboratory classes	project, presentation	The student receives a final grade based on points awarded for active participation in exercises, credit projects, and systematically submitted solutions to homework and programming tasks (0-60 points). To pass, all projects and all mandatory tasks must be submitted, and a minimum of 40 points must be obtained.

Literature

Required

1. documentation of the discussed network technologies

Additional

1. Andrew S. Tanenbaum, David J. Wetherall, Computer Networks

Operating Systems

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle studies

Form of study : full-time studies

Study profile : general academic

Obligatory status : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.140.01912.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0613 Software and applications development and analysis

USOS code : WMI.TCS.SO.OL.

Course coordinator

Jakub Kozik

Course instructor

Jakub Kozik

Period Semester 3	Form of verification of learning outcomes exam Teaching methods and hours lecture: 30 laboratory classes: 30	Number of ECTS credits 6.0

Educational aims of the course

C1	Introduction to the operating system interface defined in the POSIX standard. Developing programming skills based on this standard (POSIX programming).

C2	Understanding the basic concepts and problems related to operating system implementation, including the POSIX standard.
C3	Making students aware of the fundamental problems of concurrent programming.

Learning outcomes for the course

Code	Effects in terms of	Directional learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	the operating system interface defined in the POSIX standard.	IAN_K1_W13	written exam, project, programming tasks
W2	principles of operating system design.	IAN_K1_W13	written exam
Skills – The student can:			
U1	analyze the advantages and disadvantages of solutions used in operating system implementation.	IAN_K1_U19	written exam, project
U2	use inter-process communication mechanisms provided by the system to implement example concurrent applications.	IAN_K1_U12	programming tasks
U3	program applications based on the POSIX standard.	IAN_K1_U12, IAN_K1_U18	project, programming tasks
U4	carry out a simple programming project involving modification/extension of an example operating system.	IAN_K1_U18	project

ECTS credits balance

Form of student activity	Average number of hours* devoted to completed types of activities
lecture	30
laboratory classes	30
project preparation	60

independent solving of computer tasks	20
exam preparation	30
exam participation	2

Total student workload	Number of hours 172	ECTS credits 6.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	Basic operating system interface – POSIX standard – processes, files, signals.	W1, U3
2.	Concurrency and process synchronization mechanisms.	W2, U2
3.	Operating system architectures – monolithic systems, microkernel systems.	W2
4.	Process scheduling methods.	W2
5.	Analysis of the MINIX operating system microkernel implementation.	U1
6.	Input/output system – general resource management issues, deadlock avoidance/detection mechanisms, implementation of the input/output system in MINIX.	W2, U1
7.	Memory management – segmentation and paging mechanisms, implementation of memory and process management in MINIX.	W2, U1
8.	File system – types of disk space organization, MINIX file system, implementation of the file server in MINIX.	W2, U1
9.	Implementation of a programming project involving modification/extension of an example operating system	U4

Extended information

Teaching methods:

project method, multimedia presentation lecture, case studies, laboratory classes

Type of classes	Forms of credit	Course credit requirements
lecture	written exam	To pass the course, more than 50% of points on the written exam and a positive grade from the laboratory classes are required.

Type of classes	Forms of credit	Course credit requirements
laboratory classes	project, programming tasks	To pass the laboratory classes, timely and correct completion of two programming projects and two programming tasks is required. Activity during classes may improve the grade but does not influence passing the classes.

Prerequisites and additional requirements

- ability to program in C language
- knowledge of the user side of a UNIX family system

Literature

Required

1. Andrew S. Tanenbaum, Systemy operacyjne, Helion 2015

Additional

1. Andrew S Tanenbaum, Albert S Woodhull, Operating Systems Design and Implementation, 3rd Edition, Pearson Prentice Hall 2009.
2. POSIX.1-2017, The Open Group Base Specifications Issue 7, 2018 edition, IEEE Std 1003.1™-2017 (Revision of IEEE Std 1003.1-2008)
3. A. Silberschatz, J.L. Peterson, G. Gagne: Podstawy systemów operacyjnych. WNT

Programmer's Toolkit

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle studies

Form of study : full-time studies

Study profile : general academic

Obligatory status : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.110.03336.22

Languages of instruction : Polish

Disciplines : Computer Science

ISCED classification : 0611 Computer use

USOS code : WMI.TCS.SP.OL

Course coordinator

Grzegorz Gutowski

Course instructor

Grzegorz Gutowski

Period Semester 1	Form of verification of learning outcomes grade assessment Teaching methods and hours laboratory classes: 30	Number of ECTS credits 3.0

Educational aims of the course

C1	The student knows basic tools supporting the work of a computer scientist
C2	The student knows the principles of versioning and can use a version control system

C3	The student can navigate Linux from the command line at a basic level and write scripts to automate the work of a computer scientist
C4	The student knows how to use a debugger and write a simple "makefile"
C5	The student can use the learned tools to test programs and find errors in their own code
C6	The student can speak in an understandable language about issues covered in class and formulate questions to better understand the topic

Learning outcomes for the course

Code	Effects in terms of	Directional learning outcomes	Verification methods
Knowledge – The student knows and understands:			
W1	basic tools supporting the analytical work of a computer scientist, including the principles of versioning (version control systems)	IAN_K1_W03, IAN_K1_W15	grade assessment
Skills – The student can:			
U1	navigate Linux from the command line at a basic level and write scripts to automate the work of a computer scientist, knows how to use a debugger and write a simple "makefile"	IAN_K1_U04, IAN_K1_U05, IAN_K1_U11, IAN_K1_U12, IAN_K1_U21	grade assessment
Social competences – The student is ready to:			
K1	speak in an understandable language about issues covered in class and formulate questions to better understand the topic	IAN_K1_K01	grade assessment

ECTS credits balance

Form of student activity	Average number of hours* devoted to completed types of activities
laboratory classes	30

test preparation	10
preparation for classes	10
independent study of topics covered in class	10
independent solving of computer tasks	30

Total student workload	Number of hours 90	ECTS credits 3.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	Linux shell commands: basic operations on files and directories, displaying file contents, regular expressions, processes, tasks, system variables.	W1, U1, K1
2.	The grep, sed commands and awk language as tools for searching and editing text streams.	W1, U1, K1
3.	Scripts in Linux: handling conditional statements, loops, functions, and streams in the Bash shell.	W1, U1, K1
4.	Version control systems: Git, file updates, change tracking, communication with external repositories.	W1, K1
5.	Debugging: gdb, examples of errors.	W1, U1, K1
6.	Makefile: basic creation principles.	W1, U1, K1

Extended information

Teaching methods:

problem solving, laboratory classes

Type of classes	Forms of credit	Course credit requirements
laboratory classes	grade assessment	Credit based on point accumulation - assessment according to scale. Detailed assessment criteria are determined each time by the module instructors.

Literature

Required

1. The module has an original character, a prepared script with materials is binding.

Additional

1. Supporting literature:
    1. http://www.tutorialspoint.com/unix/index.htm
    2. http://www.tutorialspoint.com/awk/index.htm
    3. https://git-scm.com/book/pl/v1
    4. http://www.unknownroad.com/rtfm/gdbtut/gdbtoc.html
    5. http://mrbook.org/tutorials/make/
2. Additional topics:
    - http://wazniak.mimuw.edu.pl/index.php?title=%C5%9Arodowisko_programisty
    - http://hginit.com/
    - http://heather.cs.ucdavis.edu/~matloff/debug.html
    - http://valgrind.org/docs/manual/index.html
    - ftp://ftp.gust.org.pl/TeX/info/lshort/english/lshort.pdf
    - http://ece.uprm.edu/~caceros/latex/introduction.pdf

Tutorial

Course description

Basic information

Field of study : Analytical Computer Science

Path : -

Organizational unit : Faculty of Mathematics and Computer Science

Education level : first-cycle studies

Form of study : full-time studies

Study profile : general academic

Obligatory status : mandatory

Education cycle : 2022/23

Course code : UJ.WMIIANS.1200.02446.22

Languages of instruction : Polish

Course related to scientific research : Yes

Disciplines : Computer Science

ISCED classification : 0588 Natural sciences, mathematics and statistics, inter-disciplinary programmes, 0612 Database and network design and administration, 0613 Software and applications development and analysis, 0619 Information and Communication Technologies not elsewhere classified, 0688 Information and Communication Technologies (ICTs), inter-disciplinary programmes

USOS code : WMI.TCS.TUT.T

Course coordinator

Iwona Cieślik

Course instructor

Iwona Cieślik

Period Semester 6	Form of verification of learning outcomes graded credit Teaching methods and hours tutorial: 5	Number of ECTS credits 5.0

Educational aims of the course

C1	Developing skills in preparing written papers on a selected computer science topic.

Learning outcomes for the course

Code	Effects in terms of	Directional learning outcomes	Verification methods
Skills – The student can:			
U1	obtain information from documentation, professional literature (in Polish and English), integrate it, make own conclusions, analyses and interpretations	IAN_K1_U24	graded credit
U2	present computer science issues in an understandable language and prepare written studies on selected topics	IAN_K1_U02, IAN_K1_U21, IAN_K1_U22	graded credit
U3	demonstrate readiness to constantly adapt their knowledge and practical skills to changes occurring in computer science; understand the need to improve their qualifications	IAN_K1_U26	graded credit
Social competences – The student is ready to:			
K1	formulate bold questions that help better understand the assigned topic and stimulate the search for information in professional literature, the internet and scientific articles; approach critically the information found and their own conclusions; justify the results of their analyses	IAN_K1_K01, IAN_K1_K04, IAN_K1_K05	graded credit
K2	plan systematic work on a given topic; define priorities for preparing the assigned written work	IAN_K1_K02, IAN_K1_K03	graded credit

ECTS credits balance

Form of student activity	Average number of hours* devoted to completed types of activities
tutorial	5
gathering information for the assigned work	20
problem analysis	50

preparation of semester paper	60	
Total student workload	Number of hours 135	ECTS credits 5.0

* hour (lesson) means 45 minutes

Course content

No.	Course content	Learning outcomes for the course
1.	1. Independent acquisition of knowledge from professional literature and scientific articles. 2. Preparation of written work.	U1, U2, U3, K1, K2

Extended information

Teaching methods:

consultations, independent work

Type of classes	Forms of credit	Course credit requirements
tutorial	graded credit	Preparation of written work.

Prerequisites and additional requirements

Preparation of written work on a selected computer science topic.

Literature

Required

1. The student selects literature individually depending on the chosen topic of the prepared work.