COMPUTER SYSTEM ARCHITECTURE
Course Code: INFO 101
Type of course: Compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Cucu Ciprian, PhD
Seminar tutor: Cucu Ciprian, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Autumn	Grade	6

COURSE AIMS:

- Firstly it provides students with the theoretical elements necessary to understand the basic concepts regarding the architecture and functioning of personal computers.
- Secondly, it allows for the development of practical skills regarding the use of hardware and software resources, through laboratory applications.

ENTRY REQUIREMENTS: -

COURSE CONTENTS:

A. Numbering systems
B. Assembly Language
C. Architecture types
D. IBM-PC compatible hardware

TEACHING METHODS:

Lecture, Power Point presentations, conversation, exemplification.

LEARNING OUTCOMES:

Designing, selecting, and interconnecting hardware components and designing the hardware/software interface to create a computing system that meets functional, performance, energy consumption, cost, and other specific goals.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written paper – final evaluation – 50%; continuous assessment Laboratory activities portfolio – 50%.

RECOMMENDED READING:

- Andrew S. Tanenbaum, Structured Computer Organization (5th Edition). Prentice Hall, 2005.
- Jeff Duntemann. Assembly Language Step-by-step (2nd Edition): Programming with DOS and Linux .Wiley, 2000.
- http://www.cs.utexas.edu/~fussell/courses/cs352h/lectures/Lecture_1.pdf
- http://www.csee.umbc.edu/courses/undergraduate/421/spring03/slides/ch2-2.pdf
- http://people.cs.clemson.edu/~mark/330/chap1.pdf
- http://www.ece.umd.edu/~manoj/350/notes/book.pdf

COMPUTATIONAL LOGICS
Course Code: INFO 102
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Aldea Mihaela, PhD
Seminar tutor: Aldea Mihaela, PhD
Full time studies

Form of instruction	Number of teachinghours per semester	Number of teachinghours per week	Semester	Form of receiving a credit for a course	Number of ECTS creditsallocated
Class	56	4	Autumn	Grade	6

COURSE AIMS:

The discipline Computational logics aims to provide students opportunities to identify and use knowledge of the laws of human reasoning, for the purposes of mastering proper expertise and especially for their enforcement in the areas of artificial intelligence, analysis and synthesis of logic circuits, the automatic demonstration theorems, the logic programming.

ENTRY REQUIREMENTS: -

COURSE CONTENTS:

1. Propositional Logic: Logical operations, Logical equivalence of formulas, Duality law
2. Decision Problem. Perfect normal forms.
3. Propositional calculus elements: The concept of formula. True formulas
4. Deduction theorem. Rules of propositional calculus.
5. Logically equivalent formulas. Deductibility theorems. Formulas in propositional algebra and propositional calculus.
6. No contradiction and completeness of propositional calculus. Independence of propositional calculus axioms.
7. Predicate calculus: Definitions of predicates and quantifiers. Normal forms.
8. Predicate calculus formulas and axioms.
9. Noncontradiction and narrowly completeness of predicate calculus. Theorems of predicate calculus.
10. Equivalent formulas in predicate calculus. Axioms of predicate calculus.
11. Numeral: positional representation of numbers, algorithms for crossing a number from one base to another, the four operations in various numeral, numeral 2, 8, 16; characteristic elements.
12. Representation of numerical information in memory computer systems: fixed-point representation of numerical information, floating point representation of numerical information, arithmetic operations with floating point numbers, IEEE P754 Standard
13. Boolean functions and their realization: the notion of Boolean function of several variables, Boolean operations AND, OR, NOT
14. The operation of AND gate, OR gate, NOT gate circuits; Implementation of Boolean functions. Boolean functions applications

TEACHING METHODS:

Lecture, conversation, exemplification.

LEARNING OUTCOMES:

Acquiring fundamental knowledge concerning the discipline specific concepts: formal systems, judgments and sentences, modal logic elements, probability, predicate logic elements; training in problem solving skills necessary for circuit design and optimization of computer systems based on structural formulas, representing information in memory computer systems.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written paper –70%; continuous assessment – 30%.

RECOMMENDED READING:

- Michael R. Genesereth, Nils J. Nislsson, *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann Publishers, 1988
- S. Russell and P. Norvig, *Artificial Intelligence. A Modern Approach*, Prentice Hall, 1995
- Stphen G. Simson, *Mathematical Logic,* Department of Mathematics The Pennsylvania State University, University Park, State College PA 16802, 2010

IMPERATIVE/ PROCEDURAL PROGRAMMING

Course Code: INFO 103
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Ovidiu Domșa, PhD
Seminar tutor: Arpad Incze, PhD
Full time studies

Form of instruction	Number of teachinghours per semester	Number of teachinghours per week	Semester	Form of receiving a credit for a course	Number of ECTS creditsallocated
Class	70	5	Autumn	Grade	6

COURSE AIMS:

- Develop algorithmic thinking and skills for developing elementary algorithms.
- Learning basic tools for developing elementary algorithms.
- Knowledge of types of methods and data structure regarding algorithms and their development methods.
- Use of an advanced programming language for implementing the studied algorithms

ENTRY REQUIREMENTS:

No entry requirements needed.

COURSE CONTENTS:

1. General principles for structured programming and algorithm development.
2. Definitions: Algorithms. Characteristics. Structure, data and algorithmically steps.
3. Organizing data and structure in structural programming. Linear, alternative and while structures.
4. Algorithms: Elaboration, Correctness, Complexity and Testing.
5. Elementary algorithms. Switch variable values, alternative structures, While and repeat structures, vectors, mathematical quantification each/exist, Cartesian product algorithm.
6. Counting, Summary, Searching elementary algortitms.
7. Evaluation
8. Sub algorithms, defining parameters and variable transfer
9. Elementary sorting methods (Bubble sort, Selection Sort, Numbering Sort, Insertion Sort)
10. Sorting and searching algorithms complexity. Elementary algorithm methods. Intercalation.
11. Recursively algorithms. Recursively function.
12. C language. Elementary concepts. Vocabulary. Data definition. Input/output data in C.
13. Programming structure in C. Instructions IF, WHILE, DO, CASE
14. Elementary algorithms, applications.

TEACHING METHODS:

Lecture, conversation, exemplification, problem solving, documentation.

LEARNING OUTCOMES:

- acquisition of basic and specific knowledge about the concept of elementary algorithm;
- the ability to identify the applicability of the studied algorithms in real problems;
- understanding the need of using elementary methods to create algorithms when addressing problems from an algorithmic perspective;
- acquiring basic knowledge on the concept of algorithms complexity.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written exams – 50%;
Continuous assessment and laboratory practical works – 50%.

RECOMMENDED READING:

- Ovidiu Domsa, Imperative / Procedural programming, Course notes, 2013.
- Cormen T.H., Leiserson E.C., Rivest R.R., Introduction in algorithms, MIT Press, 2001.
- Dahl O.J., Dijkstra E.W., Hoare C.A.R., Structured Programing, Academic Press, 1972.
- C/C++ Programmers Bible,, Kris Jamsa, Lars Klander, 1997.

LINEAR ALGEBRA, ANALYTICAL AND DIFFERENTIAL GEOMETRY

Course Code: INFO 104

Type of course: compulsory

Language of instruction: English tutoring available for Erasmus students

Name of lecturer: Aldea Mihaela, PhD

Seminar tutor: Popa Lucian, PhD

Full time studies

Form of instruction	Number of teachinghours per semester	Number of teachinghours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	3	Autumn	Grade	4

COURSE AIMS:

- The overall objective of this discipline is the consolidation of the concepts of linear algebra studied in high school, including at the same time, elements of superior algebra and analytical geometry necessary for other educational objects.

ENTRY REQUIREMENTS:

Knowledge of high school algebra

COURSE CONTENTS:

1. Introduction. Algebraic structures
2. Matrix operations
3. Vector spaces. Euclidean spaces
4. Linear transformations
5. Eigenvectors and eigenvalues
6. Multiline algebra and tensor product. Bilinear applications, quadratic forms
7. Vectors
8. Lines and planes in space
9. Transformations
10. Conics
11. Quadrics
12. Differential geometry
13. Surfaces

TEACHING METHODS:

Lecture, conversation, exemplification.

LEARNING OUTCOMES:

After going through this course, students will acquire skills in using Linear algebra and analytical geometry to solve some problems in different areas.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written paper – 70%; continuous assessment – 30%.

RECOMMENDED READING:

- R. Horn, C. Johnson, Analiză matriceală, Editura Theta, 2006
- D. Lay, Linear algebra and its applications, Addison-Wesley Publishing, 2003
- V. V. Konev, Linear Algebra, Vector Algebra and Analytical Geometry, Tomsk Polytechnic University, 2009
- C. Udrişte, Problems in algebra, geometry and differential equations I, II, University Politehnica of Bucharest, 1992

MATHEMATICAL ANALYSIS
Course Code: INFO 105
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Prof. Daniel Breaz, PhD
Seminar tutor: Assis. Ioan-Lucian Popa, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Autumn	Grade	4

COURSE AIMS:

After browsing the course, the students will gain skills in the use of mathematical analysis for transposition of problems in various programming languages. So the discipline contributes to the formation of some general skills specific for the study domain.

ENTRY REQUIREMENTS:-

COURSE CONTENTS:

1.Strings.
1.1 Strings applications, real numbers strings, strings in metric spaces.
1.2 Calculation of string limits

2. Numerical series.
2.1 Applications to numerical series and convergence criteria for series with random terms.
2.2 Applications to absolute convergent series, semiconvergent series, and series with positive terms.

3. Functions between metrical spaces.
3.1 Applications regarding function calculation of the limits in one point.
3.2 Continuity of functions between metric spaces.

4. Integration of real functions.
4.1 Calculation of some integrals out of real functions.
4.2 Applications to calculate defined integrals.

5. Strings and series of functions
5.1 Applications of strings and series of functions.
5.2 Applications of rise series and Taylor series.

6. Functions derivations of more than one variable
6.1 Applications to function derivations of more than one variable, partial derivations.
6.2 Applications to functions differentials of more than one variable and functions extremes of more than one variables.
6.3 Conditioned extremes.

7. Basic knowledge regarding integrals
7.1 Improper integrals applications.
7.2 Applications of integrals with parameters.
7.3 Applications of Eulerian integrals and double integrals

TEACHING METHODS:
Lecture, discussion, exemplification.

LEARNING OUTCOMES:

In order to obtain credits for this discipline the student have to know how to work with elementary mathematical analysis notions, which are necessary in the basic theoretical bases of computer science and formal models.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Final evaluation – 50%; continuous assessment – 50%.

RECOMMENDED READING:

- Breaz D., Acu, M., Mathematic Analysis , Editura Risoprint, Cluj Napoca, 2008
- 2. Mangatiana A. Robdera, A Concise Approach to Mathematical Analysis, Springer 2003
- Graeme L. Cohen, A course in modern analysis and its applications, Cambridge University Press 2003
- Niels Jacob, Kristian P Evans, A Course in Analysis - Volume I: Introductory Calculus, Analysis of Functions of One Real Variable World Scientific Publishing 2016

DATA STRUCTURES AND ALGORITHMS
Course code: INFO 109
Type of course: compulsory
Language of instruction: English/Romanian
Name of lecturer: Corina Rotar, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Summer	Grade	6

COURSE AIMS:
Algorithms and Data Structure is a fundamental discipline which is required in the curricula of Computer Science specialization. Course content is designed for training the algorithmic thinking of the students.
Objectives:

- Develop students' ability to design software that is dedicated to solving medium complexity problems.
- Deepening the concept of data structure and gaining the skills to design abstract data types and associated libraries.
- Creating a rigorous and efficient programming style
- Developing students' ability to effectively manage information by using abstract data types and rigorously designing the algorithms to process the data.
- Drawing a coherent documentation on the applications of average complexity.

ENTRY REQUIREMENTS:

- Imperative and Procedural Programming

COURSE CONTENTS:

1. Introduction. Programming paradigms
2. Data structures. Abstract data type (ADT). Examples: Rational ADT, Compex ADT- 2 sessions
3. Dynamic memory allocation
4. Simple linked lists, circulars, stack, and queue.
5. Double Linked lists
6. ADT Trees
7. ADT tables
8. TAD Graphs.
9. Algorithms on graphs.
10. Programming methods. Divide et Impera technique.
11. Greedy method.
12. Branch and Bound method.
13. Backtracking method. - 2 sessions
14. Dynamic programming method.

TEACHING METHODS:

- Lecture, Cooperative learning, Discussion and survey, Team-based learning.

LEARNING OUTCOMES:

- Implementation and documentation of the software units in high-level programming languages and efficiently used programming environments.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

- Final evaluation (writen exam) 60%
- *Laboratory activities portfolio -40%*

RECOMMENDED READING:

- Knuth, Donald E. , *The Art of Computer Programming*, Vol I si II, Teora, 1999.
- Eckel, Bruce. Thinking in C++,
- Rotar, Corina, Algorithms and Data Structures, Lecture notes (seria Didactica)

MATHEMATICAL BASES OF COMPUTERS
Course Code: INFO 110
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Boca Maria Loredana, PhD
Seminar tutor: Boca Maria Loredana, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Summer	Grade	6

COURSE AIMS:
Acquiring fundamental knowledge in computer operation, understanding how interconnected computer components and how is the interaction human-computer.

ENTRY REQUIREMENTS:
- Fundamental knowledge in computer operation.

COURSE CONTENTS:
The course covers the following main topics:
- Logical combinational circuits
- Normal shapes, diagrams, minimization
- Numerical system: binary, octal, hexadecimal, decimal
- Encoding binary-decimal systems
- Decimal-binary and Binary-decimal conversions
- Arithmetic operations
- Semisum and sum
- Sequential logical circuits: bitable (RS, D, T, JK – Master-Slave)
- Shift registers, numbering machine
- Accumulators, multiplexers
- Finite automata

TEACHING METHODS:
Lecture, conversation, exemplification, exercises.

LEARNING OUTCOMES:
- C4.1 Defining of basic concepts and principles of computer science and mathematical theories and models.
- C4.2Interpretation of mathematical and informatics models (formal).
- C4.3 Identifying the adequate models and methods to solve real problems.
- C4.4 Using simulation to study the behavior patterns made and performance evaluation.
- C4.5 Using of formal models in specific applications for various fields.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:
Projects/Assignments –60%; continuous assessment – 40%.

RECOMMENDED READING:

- M. Ben-Ari: Mathematical Logic for Computer Science, Ed. Springer, 2001.
- M.Fitting: First-order logic and Automated Theorem Proving, Ed.Springer Verlag, 1990
- A.Thayse: From standard logic to Logic Programming, Ed. JWiley, vol1(1989), vol3(1990).

OPERATION SYSTEMS
Course Code: INFO 111
Type of course: Compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Cucu Ciprian, PhD
Seminar tutor: Cucu Ciprian, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Summer	Grade	6

COURSE AIMS:

- Developing fundametal knowledge regarding operating system concepts
- Using and configuring operating systems, focusing on Linux.

ENTRY REQUIREMENTS:

-

COURSE CONTENTS:

A. History of operating systems
B. General concepts
C. Introduction to Linux
D. Filesystems

TEACHING METHODS:

Lecture, Power Point presentations, conversation, exemplification.

LEARNING OUTCOMES:

Configuring and using various Operating Systems.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written paper – final evaluation – 50%; continuous assessment Laboratory activities portfolio – 50%.

RECOMMENDED READING:

- Tanenbaum, A.S.; Bos, H. – "Modern Operating Systems, Fourth Edition." *Pearson*, 2015.
- Silberschatz, A.; Galvin, P.B.; Gagne, G. – "Operating System Concepts Essentials", *John Wiley & Sons*, 2011.

GRAPHS ALGORITHMS
Course Code: INFO 112
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Dorin Wainberg, PhD
Seminar tutor: Dorin Wainberg, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Summer	Grade	5

COURSE AIMS:

Our aims in this course are twofold. First, to discuss some of the major results of graph theory, and to provide an introduction to the language, methods and terminology of the subject. Second, to emphasize various approaches (algorithmic, probabilistic, etc.) that have proved fruitful in modern graph theory: these modes of thinking about the subject have also proved successful in areas of informatics, and we hope that students will find the techniques learnt in this course to be useful in their future works.

ENTRY REQUIREMENTS:

Linear Algebra

COURSE CONTENTS:

1. Preliminaries. General notions.
2. Basic concepts in Graph Theory. Cyclomatic number
3. Graph traversal. Breadth First Traversal. Depth First Traversal
4. Minimum distances in graphs
5. Connected components
6. Bipartite graphs. Maximum matching problem in a bipartite graph
7. Hamiltonian paths and circuits . Chen algorithm. Foulkes algorithm. Kaufmann algorithm
8. Flow networks. Bellman-Kalaba algorithm. Ford algorithm. Dijkstra algorithm
9. Maximum flow in transport networks
10. Trees. Deffinitions and theorems.
11. Traversal of a dirrected tree
12. Trees of minimum values. Kruskal algorithm. Sollin algorithm
13. Binary trees
14. Structural trees

TEACHING METHODS:

Lecture, conversation, exemplification.

LEARNING OUTCOMES:

Modelling and solving some medium complexity level problems, using the mathematical and computer sciences knoweledges.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written paper 50%; mid-term test 30%; seminar activities 20%.

RECOMMENDED READING:

- Gross, J.L., Yellen, J., *Graph Theory and its Applications*, CRC Press LLC, 1998
- Diestel R., *Graph Theory*, Springer-Verlag, 1997
- West, D.B., *Introduction to Graph Theory*, Prentice Hall 1996
- Wilson, R.J., *Introduction to Graph Theory*, Addison Wesley Longman, 1996
- Behzad, M., Chartrand, G., Lesniak-Foster, L., *Graphs and digraphs*, Prindle, Weber and Schmidt, Boston, Massachusetts, 1979.
- Bollobas, B., *Graph theory. An introductory course*, Springer-Verlag, New York, Heidelberg, Berlin, 1979.

PROBABILISTIC AND MATHEMATICAL STATISTICS
Course Code: INFO 113
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Lucia Căbulea, PhD
Seminar tutor: Dorin Wainberg, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	42	3	Summer	Grade	5

COURSE AIMS:

This course is designed to introduce students to various topics in probability and uncertainty that they will encounter in Computer Science theory. The concepts are illustrated with actual examples from the specialized literature. Exercises are designed to encourage the student to begin thinking about probability within a theoretical context. Today, the theory of probability has found many applications in science and engineering. In this course, the students will learn the basic terminology and concepts of probability theory and statistics.

ENTRY REQUIREMENTS:

Linear Algebra

COURSE CONTENTS:

1. Field of events
2. Probability field
3. Rules for assigning and calculating probabilities
4. Classical probability distributions
5. Discret random variables
6. Continuous random variables
7. Numerical characteristics of random variables
8.The characteristic function. Moment generating function
9. The law of large numbers for random variables. Limit theorems
10. Statistical selection theory
11. Glivenko's theorem. Kolmogorov's theorem
12. Estimation theory
13. Confidence intervals
14. Statistical hypothesis testing

TEACHING METHODS:

Lecture, conversation, exemplification.

LEARNING OUTCOMES:

Modelling and solving some medium complexity level problems, using the mathematical and economics knoweledges.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written paper 50%; mid-term test 30%; seminar activities 20%.

RECOMMENDED READING:

- Micula, S., *Probability and Statistics for Computational Sciences*, Cluj University Press, 2009
- Agratini, O., Blaga, P., Coman, Gh., *Lectures on Wavelets, Numerical Methods and Statistics*, Casa Cartii de Stiinta, Cluj-Napoca, 2005.
- Feller, W., *An introduction to probability theory and its applications*, Vol.I-II, John Wiley, New York, 1957, 1966.
- Lisei, H., Micula, S., Soos, A., *Probability Theory trough Problems and Applications*, Cluj University Press, 2006.
- Milton, J.S., Arnold, J. C., *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences*, McGraw-Hill, New York, 1995.
- Shiryaev, A.N., *Probability*, Springer, New York, 1995.

DATABASES
Course Code: INFO 201
Type of course: fundamental
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Olteanu Emil, PhD
Seminar tutor: Muntean Maria, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Autumn	Grade	5

COURSE AIMS:
- Database design;
- Database normalization;
- Implementing databases, tables, primary and foreign keys and foreign key relationships;
- Modifying structure of tables;
- Adding, modifying and deleting table data;
- Querying databases.

ENTRY REQUIREMENTS: Introduction to Databases.

COURSE CONTENTS:
1. Database Architecture
2. Data models
3. Relational database
4. Relational algebra
5. Relational keys
6. Database normalization
    - The First Normal Boyce-Codd Form (1NF). Steps and examples.
    - The Second Normal Boyce-Codd Form (2NF). Steps and examples.
    - The Third Normal Boyce-Codd Form (3NF). Steps and examples.
7. SQL Select
8. Relational database schemes
9. Conceptual design of databases
    - Logical database design

TEACHING METHODS:
Lecture, Power Point presentations, conversation, exemplification.

LEARNING OUTCOMES:
- Organizing data in databases.
- Querying databases.
- Development of various database related projects.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:
Written paper – interpretative essay – 50%; continuous assessment Laboratory activities portfolio – 50%.

RECOMMENDED READING:

- Colin Titchie, Relational Database Principle, 2nd edition, UK by TJ International, Padslow, Cornwall, 2004.
- Terry Halpin, Tony Morgan, Information Modeling and Relational Databases, second edition, Morgan Kaufmann Publishers is an imprint of Elsevier. 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA.
- Mark Levene and George Loizou, A Guided Tour of Relational Databases and Beyond, Springer-Verlag Berlin Heidelberg.
- Kroenke, David M, Database Processing: Fundamentals, Design & Implementation, New Jersey: Prentice Hall, 2000.
- Saeed K. Rahimi, Frank S. Haug, Distributed Database Management Systems: A Practical Approach, Hoboken, New Jersey: Wiley Publishing INC, 2010.
- Lambert M. Surhone, Mariam T. Tennoe, Susan F. Henssonow, Distributed Database: Database Management System, Computer Storage, Routing Protocol, Beau Bassin, Mauritius: Betascript Publishing, 2010.
- Weinberg, P., Groff, J., Oppel, A., SQL The Complete Reference, Third Edition, The McGraw-Hill Companies, Inc., ISBN: 978-0-07-159255-0, 2010.
- Schneider, R., D., MySQL Database Design and Tuning, Sams Publishing, ISBN: 0-672-32765-1, 2005.
- Date, C., J., SQL and Relational Theory, 1st Edition, O'Reilly Media, Inc., ISBN: 978-0-596-52306-0, 2009.

FUNDAMENTAL ALGORITHMS
Course Code: INFO 202
Type of course: optional
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Ovidiu Domșa, PhD
Seminar tutor: Adriana Bîrluțiu, PhD
Full time studies

Form of instruction	Number of teachinghours per semester	Number of teachinghours per week	Semester	Form of receiving a credit for a course	Number of ECTS creditsallocated
Class	56	4	Autumn	Grade	6

COURSE AIMS:
- Develop algorithmic thinking and skills for developing complex algorithms.
- Learning basic tools for developing fundamental algorithms.
- Knowledge of different types of fundamental algorithms and their development methods.
- Use of an advanced programming language for implementing the studied algorithms.

ENTRY REQUIREMENTS:
Imperative and procedural programming
Algorithms and data structures
Graph algorithms

COURSE CONTENTS:
- General principles for algorithm development.
- Complexity of algorithms. Asymptotic analysis of worst case scenario.
- Numerical algorithms. Optimization of numerical algorithms. Primality. Bell numbers. Stirling numbers. Catalan numbers. Numbers with special properties.
- Sorting: HeapSort, QuickSort, RadixSort, Median-Algorithms, Lower Bounds.
- Analysis of sorting and searching algorithms complexity.
- Parallel sorting: enumeration sort, odd-even transposition sort.
- Parallel sorting: bitonic sort, quicksort on a hypercube.
- Binary search trees.
- AVL trees. Red-black trees. B-trees.
- Hash tables. Collision resolution. Hash functions.
- Graph algorithms: Transitive Closure, Shortest Path Problems, Minimum Spanning Trees.
- Branch&Bound algorithms. Exemples of problems solved with the Branch&Bound method.
- NP-complete algorithms.
- Analysis, evaluation, and feed-back.

TEACHING METHODS:
Lecture, conversation, exemplification, problem solving, documentation.

LEARNING OUTCOMES:
- acquisition of basic and specific knowledge about the concept of fundamental algorithms;
- the ability to identify the applicability of the studied algorithms in real problems;
- understanding the need of using advanced methods to create efficient algorithms when addressing problems from an specific domain;
- Acquiring advanced knowledge of algorithms complexity and apply efficient methods to solve different practical problems.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:
Written exams – 50%;
Continuous assessment and laboratory practical works – 50%.

RECOMMENDED READING:

- Adriana Bîrluțiu, Maria Muntean, Ovidiu Domsa, Fundamental Algorithms, Course notes and applications, Seria Didactică, 2015.
- Cormen T.H., Leiserson E.C., Rivest R.R., Introduction in algorithms, MIT Press, 2001.
- Dahl O.J., Dijkstra E.W., Hoare C.A.R., Structured Programing, Academic Press, 1972.
- C/C++ Programmers Bible,, Kris Jamsa, Lars Klander, 1997.

- Donald E. Knuth, The Art of Computer Programming, Volumes 1–3, Addison-Wesley Professional Volume 1: Fundamental Algorithms (3rd edition), 1997. Addison-Wesley Professional, Volume 2: Seminumerical Algorithms (3rd Edition), 1997. Addison-Wesley Professional, Volume 3: Sorting and Searching (2nd Edition), 1998. Addison-Wesley Professional.

COMPUTER_NETWORKS
Course code: INFO 203
Type of course: compulsory
Language of instruction: English
Name of lecturer: Emilian CEUCA, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Autumn	Grade	4

COURSE AIMS:
• Definition, description and explanation of key concepts, theories, methods specific to the study of Computer Networks;
• Using concepts specific to the field in order to explain the fundamental phenomena specific to the field;
• Application of principles, basic rules for understanding a written / oral or to express in writing/ orally in an appropriate manner, respectively, taking into account all elements involved in the field

ENTRY REQUIREMENTS:

COURSE CONTENTS:

1. Introduction. Classification of Computer Networks
2. Protocols. Network topologies
3. Standards. The need standardization
4. The ISO-OSI
5. The TCP / IP
6. Comparison of the OSI and TCP
7. The TCP / IP. Illustration of the case. The network protocols and IEEE
8. transport data over a data link
9. network operating systems
10. Subnetworks. design subnets
11. Applications of VLAN
12. Data protection against errors
13 Networks without wires
14 Recap. Presenting a subject exam

TEACHING METHODS:
Elicitation, Cooperative learning, Discussion and survey, Team-based learning, Active learning systems, Active listening.

LEARNING OUTCOMES:
- Developing some understanding of the role of logic and discourse representation as a tool in describing and of Computer Networks;
- having developed critical reading skills and ability to initiate own research.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:
Combined oral and written examination to verify the quality and correctness of information assimilated. (50%+50%).

RECOMMENDED READING:
- Cisco materials available in share folder.
- Emilian CEUCA course materials and presentations;

OBJECT ORIENTED PROGRAMMING
Course code: INFO 204
Type of course: compulsory
Language of instruction: English
Name of lecturer: Corina Rotar, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	70	4	Autumn	Grade	6

COURSE AIMS:

- Develop students' ability to design software that is dedicated to solving medium complexity problems by using object oriented paradigm.
- Deepening the concept of class and object, and gaining the skills to design classes and associated libraries.
- Creating a rigorous and efficient object oriented programming style
- Developing students' ability to effectively manage information by using classes and relations between classes.
- Drawing a coherent documentation on the applications of average-high complexity.

ENTRY REQUIREMENTS:

- Data Structures and Algorithms

COURSE CONTENTS:

1. Object-oriented programming paradigm. Basic concepts.
2. Programming with data abstraction. Features in C + +.
3. Classes and objects. Data members and methods.
4. Constructors and destructor. Copy constructor
5. Static keyword in classes.
6. friend keyword.Overloading binary operators.
7. Overloading operators (II).
8. Conversions.
9. Derived classes, base classes. Inheritance.
10. Inheritance. Multiple inheritance.
11. Virtual methods
12. Polymorphism.
13. Generic classes.
14. Exceptions. Standard Inputs-Outputs.

TEACHING METHODS:

- Lecture, Cooperative learning, Discussion and survey, Team-based learning.

LEARNING OUTCOMES:

- Implementation and documentation of the software units in an object oriented programming language and efficiently using the related concepts.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

- Final evaluation (writen exam) 60%
- *Laboratory activities portfolio -40%*

RECOMMENDED READING:

- Thinking in C++, Bruce Eckel
- Bjarne Stroustrup, The C++ Programming Language, Addison Wesley, 1997.
- H. Schildt: C++ - complete, Teora, 2000.
- Peter Muller: Introduction to Object-Oriented Programming Using C++ , electronic resources.

DIFFERENTIAL AND PARTIAL DERIVATES EQUATIONS

Course Code: INFO 205
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Aldea Mihaela, PhD
Seminar tutor: Wainberg Dorin, PhD
Full time studies

Form of instruction	Number of teachinghours per semester	Number of teachinghours per week	Semester	Form of receiving a credit for a course	Number of ECTS creditsallocated
Class	56	4	Autumn	Grade	4

COURSE AIMS:

Presentation with practical methods for solving of ordinary differential equations, systems of differential equations, higher order differential equations and with partial derivates of order 1 and 2

ENTRY REQUIREMENTS:

Mathematical Analysis

COURSE CONTENTS:

1. First order differential equations: Basic concepts. Cauchy problem.
2. Separable differential equations. Homogeneous equations.
3. Linear differential equations.
4. Bernoulli, Riccati, Lagrange, Clairaut Differential equations.
5. Exact differential equations; Solutions existence and uniqueness
6. Higher order differential equations: Cases and modalities for reduction the order of an equation; Linear differential equations with variable coefficients. Fundamental sets of solutions.
7. Method of undetermined coefficients . Differential equations with constant coefficients.
8. Systems of differential equations: Systems of first order differential equations, the equivalence with higher order differential equations. Cauchy problem.
9. The fundamental matrix of a system of first order linear differential equations with variable coefficients.
10. Systems of first order linear differential equations with constant coefficients. Matrix exponential
11. Autonomous systems
12. Partial derivates equations: Linear , homogeneousand nonhomogeneous first order partial derivates equations.
13. Second order partial derivates equations.
14. Equations of mathematical physics. Laplace equation.

TEACHING METHODS:

Lecture, conversation, exemplification.

LEARNING OUTCOMES:

Learning the basic techniques of solving differential calculus problems; knowledge and application of theorems, models, their properties and methods of work in the field of differential equations and partial derivatives.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written paper – 50%; continuous assessment – 50%.

RECOMMENDED READING:

- R. Redheffer, *Diffwerential Equations. Theory and applications*, Jones and Bartleft Publishers, Boston, 1991.
- J. C. Robinson, *An introduction to ordinary differential equations,* Cambridge University Press, Cambridge, 2004.

MATHEMATICAL SOFTWARE
Course Code: INFO 206
Type of course: optional
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Prof. Nicoleta Breaz, PhD
Seminar tutor: Lect. Adriana Bîrluțiu, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	42	3	Autumn	Grade	6

COURSE AIMS:

The general aim related to this course consists in getting knowledge which allows to initiate students in the use of mathematical software products, applied in different problems with mathematical component, arising in various fields of science and technique and in general, in the use of computer tools in an interdisciplinary context.

ENTRY REQUIREMENTS:-

COURSE CONTENTS:

I. Mathematical Software Toolboxes -general issues
1. The use of specific software in the solving of mathematical problems
2. Types of mathematical software

II. Microsoft Excel spreadsheet program
1. Editing Excel formula
2. Using of Excel predefined functions
3. Mathematical Excel functions
   3.1. Trigonometrical and mathematical functions
   3.2. Statistical functions
4. Excel statistical charts
   4.1. Creating charts
   4.2. Formatting charts
   4.3. Printing and interpretation of the charts
5. Practical applications in Excel

III. Introduction in MATLAB
1. Working with MATLAB sessions
2. Constants, variables, predefined functions, arithmetical, logical and relational operators
3. Instructions for reading, editing and assigning
4. Commands for 'script' m- file
5. Instructions for flow control, branching and efficiency evaluation
6. Functions (procedures) in MATLAB

IV. Mathematical functions in MATLAB
1. Basic functions in Matlab
   1.1. Functions for linear algebra and matriceal calculus
   1.2. Functions for elementary math and trigonometric
   1.3. Functions for data analysis
   1.4. Functions for polynomial calculus
   1.5. Functions for numerical methods
   1.6. Functions for graphics
2. Matlab specialized toolboxes
   2.1. Functions for statistics
   2.2. Functions for regression modeling
   2.3. Functions for curves fitting
   2.4. Functions for optimization
   2.5. Function for equations
3. Applications in Matlab

TEACHING METHODS:

Lecture, discussion, exemplification.

LEARNING OUTCOMES:

- Using of some software products as Excel and Matlab to solve problems that requires large and hard calculation and also to simplify the way how the results are returned;
- Developing of software components for interdisciplinary projects.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Practical project – 50%; continuous assessment – 50%.

RECOMMENDED READING:

- N.Breaz, A. Bîrluțiu, Mathematical software, theory and applications, Seria Didactică, Univ. "1 Decembrie 1918"Alba Iulia, (in printing)
- D. J. Higham, N. J. Higham, MATLAB Guide, 2nd edition, SIAM, 2005
- 3.P. Marchand, O. T. Holand – Graphics and GUI with MATLAB, 3rd edition, Barnes and Noble, 2003
- Cleve Moler – Numerical Computing in MATLAB, SIAM, 2005
- ***– Documentation for MathWorks Products, R2009a- http://www.mathworks.com/

NUMERICAL CALCULUS
Course Code: INFO 209
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Prof. Daniel Breaz, PhD
Laboratory tutor: Assis. Ioan-Lucian Popa, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Summer	Grade	5

COURSE AIMS:
Introducing basic concepts and methods of numerical analysis. Initiating students in methods of numerical programming for solving mathematical problems and for start using numerical software. Students have to know the fundamental concepts of numerical analysisand various numerical algorithms. These specific objectives allow modeling and solving complex problems using knowledge of mathematics and informatics

ENTRY REQUIREMENTS:-

COURSE CONTENTS:

1. Elements of approximation theory and matrix analysis
- 1.1 Analysis and evaluation of arithmetic expressions
- 1.2 Items of errors theory and floating point arithmetic
- 1.3 Calculating the determinant and inverse of a matrix

2. Methods and numerical algorithms. Differences calculus
- 2.1 Gauss elimination method
- 2.2 Total elimination method

3. Functions approximations
- 3.1 Cholesky method
- 3.2 Onicescu method
- 3.3 Iterative methods
- 3.4 Successive approximations method
- 3.5 Tangent method
- 3.6 Secant method

4. Numerical differention and integration algorithms
- 4.1 Bairstrov method
- 4.2 Finite differences methods
- 4.3 Divided differences methods

5. Numerical algorithms for solving algebraic equations
- 5.1 Approximation in mean square
- 5.2 Numerical differentiation

6. Items of Symbolic Calculus
- 6.1 Quadrature formulas of Gauss and Newton Cotes type
- 6.2 Numerical integration using Taylor series
- 6.3 Multipas methods

TEACHING METHODS:
Lecture, discussion, exemplification.

LEARNING OUTCOMES:
In order to obtain credits for this discipline, the students have to operate with elementary items of numerical analysis and use soft for solving different mathematical problems.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:
Final evaluation – 50%; Laboratory activities – 50%.

RECOMMENDED READING:

- Eugen K. Blum – Numerical Analysis and Computation: Theory and Practice, Addison-Wesley, 1972.
- R.L. Burden, L.J. Faires – Numerical Analysis, PWS Kent, 1986
- S. Nakamura – Numerical Analysis and Graphic Visualization in MATLAB, Pretice-Hall, 1996
- Cesar Perez Lopez, MATLAB Programming for Numerical Analysis,Apress, 2014
- William Bober, Chi-Tay Tsai, Oren Masory, Numerical and Analytical Methods with MATLAB, CRC Press, 2009

FORMAL LANGUAGES AND AUTOMATION
Course Code: INFO 210
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Boca Maria Loredana, PhD
Seminar tutor: Boca Maria Loredana, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Summer	Grade	4

COURSE AIMS:

- Acquiring fundamental knowledge on the concept of mathematical modeling, the mathematical models, deterministic scheduling and implementation of a computer language;
- Formation of skills necessary to solve complex problems by interpreting expressions and instructions of a programming language.

ENTRY REQUIREMENTS:

- Fundamental knowledge in computer operation.

COURSE CONTENTS:
The course covers the following main topics:

- mathematical principles for formal languages theory
- grammars and formal language
- finite automation accounting
- regular expressions
- grammars and regular expressions
- grammars and independent content language
- compilation theory elements

TEACHING METHODS:
Lecture, conversation, exemplification, exercises.

LEARNING OUTCOMES:
C4.1 Defining of basic concepts and principles of computer science and mathematical theories and models.
C4.2Interpretation of mathematical and informatics models (formal).
C4.3 Identifying the adequate models and methods to solve real problems.
C4.4 Using simulation to study the behavior patterns made and performance evaluation.
C4.5 Using of formal models in specific applications for various fields.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:
Projects/Assignments –60%; continuous assessment – 40%.

RECOMMENDED READING:

- NPTEL >> Computer Science and Engineering >> Theory of Automata, Formal Languages and Computation (Video) >> GRAMMARS AND NATURAL LANGUAGE PROCESSING
- Formal language, Keijo Ruohonen, 2009, http://math.tut.fi/~ruohonen/FL.pdf

WEB TECHONOLOGIES
Course Code: INFO 211
Type of course: Compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Cucu Ciprian, PhD
Seminar tutor: Cucu Ciprian, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Summer	Grade	5

COURSE AIMS:

1. Understanding major concepts regarding the World Wide Web, such as the theoretical aspects and practical implications of the client-server model
2. Developing strong web applications using up-to-date practices and tools

ENTRY REQUIREMENTS:

-

COURSE CONTENTS:

A. HTTP (HyperText Transfer Protocol) and the Web Server
B. HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets)
C. Client-side Programming Using JavaScript
D. Server-side Programming Using PHP

TEACHING METHODS:

Lecture, Power Point presentations, conversation, exemplification.

LEARNING OUTCOMES:

Students will be able to understand how things work on the Web from the technology point of view and to create interoperable and functional websites.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written paper – final evaluation – 50%; continuous assessment Laboratory activities portfolio – 50%.

RECOMMENDED READING:

- *Felke-Morris, T. - Basics of Web Design*: *HTML5 & CSS3. Addison Wesley,* Boston, 2011.

DATABASE MANAGEMENT SYSTEMS
Course Code: INFO 212
Type of course: Compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Muntean Maria-Viorela, PhD
Seminar tutor: Muntean Maria-Viorela, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	42	3	Summer	Grade	5

COURSE AIMS:
- Acquiring knowledge of design and web database management;
- Acquiring knowledge of data organization according to the requirements of web communication, specific query;
- Developing skills for dialogue between web technologies and databases;
- Developing skills in validation databases using specific Web technologies..

ENTRY REQUIREMENTS:
Databases course.

COURSE CONTENTS:
1. INTRODUCTION TO DATABASES
2. CLIENT-SERVER DATABASE STRUCTURES
2.1 Bi-dimensional databases
2.2. Redundant data in client-server applications
2.3. A comparison of client-server databases architectures
3. MODERN APPROACHES IN COLLECTING AND STRUCTURING DATA
3.1. Introduction to PHP object-oriented programming
3.2. Introduction to MySQL
3.3. PHP-MySQL database application development
3.4. The main MySQL commands
3.5. High level of application development and administration in DBMS
4. STANDARD TRANSACTIONS IN DBMS APPLICATIONS
5. SERVICE-ORIENTED ARCHITECTURE DESIGN
6. CLASSES AND COMPATIBILITES IN DESIGNING CLIENT-SERVER APPLICATIONS
7. CONFIGURATION OF CLIENT-SERVER APPLICATIONS WITH DBMS SUPPORT
7.1. Configuration of service-oriented client-server applications
7.2. Configuration of data mining oriented client-server applications
8. INFORMATION SCALABILITY
8.1 Information retrieval techniques in client-server applications
8.2 Information retrieval techniques by using JOIN method
8.3 Types of JOINS used in knowledge discovery in databases

TEACHING METHODS:
Lecture, Power Point presentations, conversation, exemplification.

LEARNING OUTCOMES:
- The identification of base concept for organizing data in databases.
- The identification and explanation of base models for the organizing and management of data in databases.
- The development of various database related projects.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:
Written paper – final evaluation – 50%; continuous assessment Laboratory activities portfolio – 50%.

RECOMMENDED READING:

- W. Jason Gilmore, Beginning PHP and MySQL From Novice to Professional, Fourth Edition, Apress, 2010, ISBN-13 (pbk): 978-1-4302-3114-1, ISBN-13 (electronic): 978-1-4302-3115-8.
- Larry Ullman, PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide (4th Edition), 2011, ISBN-10: 0321784073, ISBN-13: 978-0321784070.
- Williams E. Hugh; Lane, David – Web Database Applications with PHP and MySQL, O'Reilly and Associates, 2002.

- Janet Valade, PHP and MySQL For Dummies, 4th Edition, 2009, ISBN: 978-0-470-52758-0.
- Kroenke, David M, Database Processing: Fundamentals, Design & Implementation, New Jersey: Prentice Hall, 2000.
- Saeed K. Rahimi, Frank S. Haug, Distributed Database Management Systems: A Practical Approach, Hoboken, New Jersey: Wiley Publishing INC, 2010.
- Lambert M. Surhone, Mariam T. Tennoe, Susan F. Henssonow, Distributed Database: Database Management System, Computer Storage, Routing Protocol, Beau Bassin, Mauritius: Betascript Publishing, 2010.
- Weinberg, P., Groff, J., Oppel, A., SQL The Complete Reference, Third Edition, The McGraw-Hill Companies, Inc., ISBN: 978-0-07-159255-0, 2010.
- Graham Ian – The XHTML 1.0 Web Development Sourcebook, John Wiley and Sons, 2000.
- Shea, Dave; Holzshlag E. Molly – The Zen of CSS Design: Visual Enlightement for the Web – Peachpit Press, 2005.
- Graham, Ian; Quin, Liam – The XML Specification Guide, John Wiley and Sons, 2000.
- Danesh, Arman – Javascript in 10 Steps or Less, Wiley Publishing Inc., 2004.
- Moulding, Peter – The PHP Black Book – Paraglyph Publishing, 2002.
- Welling, Luke; Thomson Laura – Php and MySQL Web Development, Sams, 2001.
- www.w3schools.com
- www.php.net

ADVANCED PROGRAMMING TECHNIQUES
Course Code: INFO 213
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Kadar Manuella, PhD
Seminar tutor: Domsa Ovidiu, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Summer	Grade	5

COURSE AIMS:

- General objectives of the course:
- Advanced programming in Java covers programming for both single system software distribution and across networks/devices. The course provides in depth coverage of object serialization, Java Beans, Servlets, Java Server pages JSP, Java Server Faces JSF, networking, remote objects (RMI), and distributed computing through Jini. The course offers many examples and applications that will be implemented within labs.
- Specific objectives of the course
- Students will understand the advanced topics in Java programming, they will be in a position to do commercial Java development both for single services and for distributed processes across multiple devices.

ENTRY REQUIREMENTS:
Object Oriented Programming (FI207)
Algorithms and data structures (FI104)
Basic algorithms (FI203)

COURSE CONTENTS:
Course (learning units)
1.Introduction to advanced programming techniques. Divide et Impera programming technique. Dynamic programming technique. Branch and Bound technique. Greedy technique. Backtracking technique.
2.Advanced programming techniques. Sorting algorithms. Multiple execution threads. Input-output streams. Serializing data.
3.Introduction to Java programming language. History and characteristics of Java programming language. Portability and security of Java programmes execution. Java data types.
4.Java versus C++. Java versus .Net. Object-oriented programming. Classes and objects; namespaces and packages. Creating and destroying objects.
5.Inheritance and class hierarchy. Abstract classes and interfaces. Exception handling
6.Java graphical interface (AWT and Swing). Graphical components: containers and controls. Managing the position of a graphical component
7.Java Graphical User Interface (AWT and Swing). Listening and handling events generated by graphical components. Graphical contexts and drawing area (canvas).
8.Java Graphical User Interface (AWT and Swing). Dialogs and menus.
9.Java and Internet services. WEB programming.
10.Web clients: applets. Applet Methods.
11.Applet Class Loaders. Applets security issues. Web Server: servlets and JSP pages.
12.JavaBeans. Using JavaBeans components in JSP pages.
13.Java database connectivity. Database access using JDBC.
14.Java database connectivity. Execution of an SQL statement. Result processing. Closing database connections in Java.

Seminars-laboratories
1. Introduction to Netbeans 7.0 integrated development environment. Basics of Java programming language.
2. Java basic statements
3. Java classes and objects. Objects. Constructors. Class variables.
4. Java classes and objects. Static methods. Inheritance
5. Method Overriding in Java. Data hiding and encapsulation.
6. Abstract classes and methods in Java
7. Java applets. Graphical User Interfaces components.
8. Events generated by AWT components
9. Java Swing. JFrame, JApplet, JPanel, Borders
10. Java Swing.Tabbed Panes, Scrolling Panes, Split Panes
11. Java Swing. Labels and buttons

12. JList. JComboBox. JSpinner.
13. JTree Text Components. JTable. Menus. JtoolBar
14. Individual project presentation based on the knowledge acquired during courses and laboratories.

TEACHING METHODS:
The course is given as a combination of lectures and laboratories. There is a 100% attendance requirement for laboratories. Work in small groups. Compulsory assignments. Instruction is a combination of lectures, laboratories, group work and individual work.

LEARNING OUTCOMES:
Professional competences
C1. Programming in high-level languages
C1.1 The appropriate description of programming paradigms and of specific language mechanisms, as well as the identification of differences between semantic and syntactic aspects.
C1.2 The explaining of existing software applications using different abstraction layers (architecture, packages, classes, methods), correctly using base knowledge.
C1.3 The development of correct source codes and the testing of various components in a known programming language, given a set of design specifications.
C2. Development and maintenance of computer applications
C2.1 The identification of appropriate methodologies for software systems development.
C2.2 The identification and explanation of appropriate mechanisms for software systems specification.
C2.3 The use of methodologies, specification mechanisms and development environments for the development of computer applications.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:
A two-hour written examination (60% of the final grade)
Laboratory activities portfolio (40% of the final grade)

RECOMMENDED READING:

- ECKEL, Bruce, Thinking in Java, 4th ed., Upper Saddle River, New JerseyUpper Saddle River, New Jersey : Prentice Hall : Pearson Education, 2006, ISBN 0-13-187248-6, 978-0-187248-6.
- Bruce Eckel, Thinking in Java, (3rd edition), http://www.bruceeckel.com.
- Marty Hall, Core Servlets and JavaServer Pages, http://coreservlets.com
- POPESCU, Nirvana, Data structures and algorithms using Java / Nirvana Popescu, Bucureşti : Politehnica Press, 2008, ISBN 978-973-7838-62-9.
- HAROLD, Elliotte Rusty, Java Network programming, Sebastopol, CA: O'Reilly, 2005, ISBN 978-0-596-00721-8.
- SCHILDT, Herbert, Java: a beginner's guide, 4th ed., New York : McGraw-Hill, 2007, ISBN 978-0-07-226384-8.
- BELL, Douglas, PARR, Mike, Java for students, Harlow, England : Prentice Hall, 2010, ISBN 978-0-273-73122-1.
- BARNES, David J., Object-oriented programming with Java: an introduction, NEW JERSEY: PRENTICE HALL, 2000.
- DUDNEY, Bill, LEHR, Jonathan, WILLIS, Bill, MATTINGLY, LeRoy – Mastering JavaServer Faces, 2004 by Wiley Publishing Inc., Indianapolis, Indiana.
- GEARY, David M. - Core JavaServer faces, David Geary, Cay Horstmann.—2nd ed., 2007 Sun Microsystems, Inc., Network Circle, Santa Clara, California.
- IANG, Y. Daniel - NetBeans Tutorial - For Introduction to Java Programming, 2005.
- SINCLAIR, Joseph - Java Web Magic, Macmillan Computer Publishing, 1997.
- TIDWELL, Doug - Tutorial: XML programming in Java, Cyber Evangelist, developerWorks XML Team, 1999.
- PELEGRI-LLOPART, Eduardo, Cable, Laurence P. G. - How to be a Good Bean, 1997 by Sun Microsystems Inc., San Antonio Road, Palo Alto, CA.
- PELEGRÍ-LLOPART, Eduardo, Cable, Larry - JavaServer Pages Specification ("Specification"), 1999 Sun Microsystems, Inc., Palo Alto, CA.
- BERGSTEN, Hans - JavaServer Faces, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA.
- *** JDK Documentation, http://java.sun.com.
- http://www.developer.com/java/data/
- http://www.moreservlets.com/
- http://myfaces.apache.org/
- http://www.java2s.com/
- http://java.sun.com/docs/books/tutorial/getStarted/cupojava/netbeans.html#netbeans
- http://www.netbeans.org/

OPTIMIZATION TECHNIQUES
Course Code: INFO 214
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Aldea Mihaela, PhD
Seminar tutor: Aldea Mihaela, PhD
Full time studies

Form of instruction	Number of teachinghours per semester	Number of teachinghours per week	Semester	Form of receiving a credit for a course	Number of ECTS creditsallocated
Class	42	4	Summer	Grade	4

COURSE AIMS:

First, discipline aims, learning to analyze and decide logically and rigorously. On the other hand, it contributes to a multidisciplinary preparation of future IT specialists, aiming in this way to familiarize students with the concepts and techniques of mathematical modeling of social and economic phenomena.

ENTRY REQUIREMENTS:

Linear Algebra

COURSE CONTENTS:

1. Solving a linear programming problem by graphical and algebraic methods\
2. Simplex method for solving linear programming problems
3. Duality. The dual simplex algorithm
4. Reoptimization of linear programming problems
5. Parametric linear programming
6. Transport problems.
7. Reoptimization of transport problems.
8. Parametric transport problems.
9. Special transport problem.
10. Integer linear programming – Gomory methods
11. Dantzig-Manne algorithm for solving integer linear programming problems.
12. Bellman method
13. Enumeration and evaluation methods.

TEACHING METHODS:

Lecture, conversation, exemplification.

LEARNING OUTCOMES:

Knowing the mathematical basic elements of optimization algorithms, familiarity with the use of optimization techniques and algorithms to solve problems.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written paper – 50%; continuous assessment – 50%.

RECOMMENDED READING:

- G. David – *Linear and Non Linear Programming,* Addison Wesley, Massachusetts, 1989.
- G. L. Nemhauser, L. A. Wolsey – *Integer and combinatorial optimization*, John Wiley & Sons Inc, New York, 1999.

GEOMETRICAL MODELING AND COMPUTER GRAPHICS
Course code: INFO 301
Geometrical Modeling And Computer Graphics (2016-2017)
Type of course: compulsory
Language of instruction: English
Name of lecturer: Emilian CEUCA, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Autumn	Grade	6

COURSE AIMS:
• Definition, description and explanation of key concepts, theories, methods specific to the study of Geometrical Modelling And Computer Graphics;
• Using concepts specific to the field in order to explain the fundamental algorithms, technics of image acquisition and processing
• Application of principles, basic rules for using equations and system to perform applications in Computer Graphics

ENTRY REQUIREMENTS:

COURSE CONTENTS:
1. Introduction. General Terms and Concepts;
2. Modelling and representation. Using 3D graphics;
3. Programming languages and equipment tools. 3D object manipulation;
4. Spatial Transformations, plane coordinate systems and two spherical coordinates.
5. Highlights of Cartesian coordinates attached observer;
6. 2D Transformations;

TEACHING METHODS:
Elicitation, Cooperative learning, Discussion and survey, Team-based learning, Active learning systems, Active listening.

LEARNING OUTCOMES:
- Developing some understanding of the role of logic and discourse representation as a tool in describing and analysing algorithms and data processing for Geometrical Modelling And Computer Graphics ;
- Having developed critical reading skills and ability to initiate own research.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:
Combined oral and written examination to verify the quality and correctness of information assimilated. (50%+50%).

RECOMMENDED READING:
- EMILIAN CEUCA – Geometrical Modelling And Computer Graphics, course materials
- Open CV Library – public internet Library

SYSTEM ANALYSIS AND DESIGN

Course Code: INFO 302
Type of course: Compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Hutanu Constantin, PhD
Laboratory tutor: Boca Loredana, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	48	4	Autumn	Grade	6

COURSE AIMS:

- This course explaining of existing software applications using different abstraction layers (architecture, packages, classes, methods), correctly using base knowledge.
- The main objective of this course is the development of communication and collaboration abilities for IT&C solutions and service projects design and the identification and explanation of base computer models that are suitable for the application domain.

ENTRY REQUIREMENTS:

Algorithms and data structures.
Computational logics.
Object-oriented programming.

COURSE CONTENTS:

1. Introduction in system analysis and design
1.1. Programs, applications, systems
1.2. Informatic technology professions
1.3. Organizing the activity within an organization
1.4. Presentation of abstract data types
2. Program engineering. Soft development paradigms
2.1. Lyfe cycle of a program
2.2. Definition of proram engineering
2.3. Clasical life cycle
2.4. 4th generation techniques. Paradigm combinations. Program engineering.
3. Scheduling software projects.
3.1. Human factors involved in program developping.
3.2. Organizing the process of making the program
3.3. Other scheduling activities. Software project planning.
4. Analysis and design techniques
4.1. Analysis of requirements
4.2. Communication techniques
4.3. Principles of analysis. Analyzing methods classification
4.4. Defining the requirements. Reviewing the definitions.
5. Reference models in system design
5.1. Conceptual data model
5.2. Comceptual data modelling using the E-R model
5.3. Generalization. Application specific rules.
5.4. Data modelling steps. Examples
6. Essential processing model
6.1. Data cycle diagram
6.2. Modelling data processing in the development cycle of an application
6.3. DFD examples
6.4. Step-by-step approach in processing models
7. Fundamental elements in system design
7.1. Fundamentals in design
7.2. Design steps. Design concepts.
7.3. Architecture design.
7.4. Design documentation
8. Logical data design.
8.1. Logical data model. Relational model.

8.2. Normalization concepts. Normal forms.
8.3. Transforming E-R diagrams into relations.
8.4. Obtaining the logical data model. Events analysis.
8. Logical data design.
8.1. Logical data model. Relational model.
8.2. Normalization concepts. Normal forms.
8.3. Transforming E-R diagrams into relations.
8.4. Obtaining the logical data model. Events analysis.
9. Processing design
9.1. General design decisions
9.2. Implementation models
9.3. General processes design
9.4. Primary processes design
9.5. Tools for proces defining
10. Physical data design
10.1. Physiscal data design process
10.2. Analysis of data volume and usage
10.3. Data distribution strategies
10.4. Folder organizing. Indexing.
10.5. Integrity requirements
11. Interface design
11.1. Interface design targets/duties
11.2. Manual, batch and online processing
11.3. Designing entry documents and reports
11.4. Human-computer interaction
12. System implementation and testing
12.1. Program specifications
12.2. Testing
12.3. Installing and maintaing systems
12.4. Using and maintaining systems.

TEACHING METHODS:

Lecture, conversation, exemplification.

LEARNING OUTCOMES:

- The use of appropriate criteria and methods for the evaluation of computer applications;
- The development of dedicated computer projects;
- Forming the student ability of analyzing and designing informatics systems: the conception, design, implementation and maintenance of informatics systems and programs, along with the necessary technical documentation;
- Leading IT&C solutiona projects, assuring the good functioning,monitoring and developing implemented IT&C solutions , asigurarea funcţionalităţii, monitorizarea şi dezvoltarea soluţiilor IT&C implementate, training staff to use implemented IT&C solutions, coordination of expert teams;
- Designing projects for IT&C solutions and services, designing/redesigning projectsfor the most complex system components, coordination of IT&C projects and teams, monitoring the perfomance of implemeted IT&C solutions, training the staff to use IT&C.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written exam – 50%; continuous assessment – 50%.

RECOMMENDED READING:

- Lungu, I., Sabău, Gh., Velicanu, M. - Sisteme informatice. Analiză, proiectare şi implementare, Ed. Economică. 2003;
- Alexandrescu, A. – Programarea modernă în C++: Programare generică şi modele de proiect aplicate, Ed. Teora, Bucureşti, 2005.
- Jeffrey A. Hoffer, Joey F.George - Modern Systems Analysis and Design. Second Edition, Ed. Addison Wesley Longman, New York, Sydney, 1999;
- Popescu, Elena; Popescu, Gh., - Elaborarea sistemelor informatice în contextul informatic actual, Constanţa Ovidius University Press, 2001

ARTIFICIAL INTELLIGENCE
Course Code: INFO 304
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Ioan Ileana, PhD
Seminar tutor: Maria Muntean, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Autumn	Grade	6

COURSE AIMS:

- The course is a coherent introduction in Artificial Intelligence area, including theoretical and practical approaches.
- The identification of appropriate models and methods for solving real-life problems.
- The use of methodologies, specification mechanisms and development environments for the development of computer applications.
- The use of computer and mathematical models and tools to solve specific problems in the application field.

ENTRY REQUIREMENTS:

-

COURSE CONTENTS:

- Introduction. Ai definitions. Short hystory of ai. Ai components
- Problem solving. Solving problems by searching. Uninformed search strategies. Informed (heuristic) search strategies
- Other problem solving strategies. Constraint satisfaction problems. Adversarial search (games)
- Knowledge representation
- Knowledge representation by rules
- Structured knowledge
- Uncertain knowledge and reasoning (fuzzy)
- Planning and learning in AI systems
- Artificial neural networks (ANN) foundations
- ANNs applications
- Expert Systems foundations
- Intelligent agents and robots.

TEACHING METHODS:

Lecture, conversation, exemplification.

LEARNING OUTCOMES:

- The use of methodologies, specification mechanisms and development environments for the development of computer applications.
- The identification and explanation of base computer models that are suitable for the application domain.
- The use of computer and mathematical models and tools to solve specific problems in the application field.
- The identification of appropriate models and methods for solving real-life problems.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written exam - 40%; continuous assessment (laboratory) - 40%, final test - 20%.

RECOMMENDED READING:

- Ioan Ileană, Corina Rotar, Maria Muntean, *Inteligenţă artificială*, Editura Aeternitas, 2009.
- RUSSELL, Stuart J., NORVIG, Peter, *Artificial Intelligence: a modern approach, 3rd ed*., Upper Saddle River, NJ: Pearson Education, 2010,
- NILSON, N. J. - *Artificial Intelligence*. A New Synthesis, Kaufmann Pbs., 1998

MULTIMEDIA TECHNIQUES AND TECHNOLOGIES
Course Code: INFO 306
Type of course: elective
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Kadar Manuella, PhD
Seminar tutor:Incze Arpad, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	56	4	Autumn	Grade	6

COURSE AIMS:
General objectives of the course:

- The Multimedia Techniques and Technologies course presents scientific and technical principles of media capture and computer representation. It is focused on methods of operation and application of computer software and systems that enable delivery of multimedia productions and also contains descriptions and examples of methods used for compression of symbolic data, as well as audio, image and video data.
- Data compression is discussed taking into consideration novel ways of data representation in order to take very little storage, with the possibility of reconstruction of the original data from the compressed version. The course offers many examples and applications such as: examples in Matlab, Adobe Flash, Adobe PhotoShop to be solved within the labs.

Specific objectives of the course

By taking this course the students will be able to:
• understand various concepts associated with multimedia technology and computing
• understand the components of multimedia systems
• explain some desirable features for multimedia systems
• explain the basic concepts of multimedia elements' representation
• implement and discuss various compression techniques
• explain how a compression system works
• analyse the advantages and disadvantages of data compression.

ENTRY REQUIREMENTS: -;

COURSE CONTENTS:
Course (learning units)

1. Introduction to multimedia technology
2. Color model and human vision. Color spaces
3. Data compression. Compression techniques and algorithms
4. Multimedia data compression standards
5. Image and sound
6. Video frames. Video frames digitization and compression
7. Audio data representation and processing. Audio compression
8. Semantic annotation of images
9. Video segmentation
10. Multimedia society - where are we going?

Seminars-laboratories
1. Introduction to MATLAB programming environment
2. MATLAB functions
3. MATLAB arrays
4. MATLAB graphics
5. Image processing using MATLAB
6. Image compression using MATLAB
7. The design and implementation of image compression techniques using MATLAB
8. The design and implementation of audio compression techniques using MATLAB
9. The design and implementation of video compression techniques using MATLAB
10. Project presentation and evaluation.**TEACHING METHODS:**

The course is given as a combination of lectures and laboratories. There is a 100% attendance requirement for laboratories. Work in small groups. Compulsory assignments. Instruction is a combination of lectures, laboratories, group work and individual work.

LEARNING OUTCOMES:
Professional competences
C2. Development and maintenance of computer applications
C2.4. Use of appropriate criteria and methods for the evaluation of computer applications.
C2.5. Development of dedicated computer projects.
C3. Use of computer tools in an interdisciplinary context
C3.1. Description of concepts, theories and models used in the application field.
C3.2 Identification and explanation of base computer models that are suitable for the application domain.
C3.3. Use of computer and mathematical models and tools to solve specific problems in the application field.
C3.4. Data and model analysis.
C3.5. Development of software components of interdisciplinary projects.
Transversal competences
CT3. Use of efficient methods and techniques for learning, scientific inquiry and development of the capacities of using knowledge, of adapting to a dynamic society and of communication in English.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:
A two-hour written examination (60% of the final grade)
Laboratory activities portfolio (40% of the final grade)

RECOMMENDED READING:

- VAUGHAN, Tay, Multimedia: making it work, 7th ed., New York, San Francisco, Chicago: McGraw-Hill: McGraw-Hill: McGraw-Hill, 2008, 0-07-226451-9, 978-0-07-226451-7.
- WEIXEL, Suzanne, FULTON, Jennifer, BARKSDALE, Karl, MORSE, Cheryl, MORSE, Bryan, Multimedia basics, Boston, Mass. : Course Technology, 2004, ISBN 0-619-05533-2, 978-0-619-05533-2
- STEINMETZ, Ralf, Multimedia applications, Berlin, Heidelberg: Springer: Springer, 2004, ISBN 3-540-40849-5, 978-3-540-40849-9.
- CHOU, Philip A, SCHAAR, Mihaela van der, Multimedia over IP and wireless networks: compression, networking, and systems, Burlington, MA : Academic Press, 2007, ISBN 0-12-088480-1, 979-0-12-088480-3.
- OHM, Jens-Raine, Multimedia communication technology: representation, transmission and identification of multimedia signals, Berlin, Heidelberg: Springer : Springer, 2004, ISBN 3-540-01249-4, 978-3-540-01249-8.
- GARRAND, Timothy, Writing for multimedia and the Web: a practical guide to content development for interactive media, 3rd ed., Burlington, MA : Focal Press, 2006, ISBN 0-240-80822-3, 978-0-240-80822-2.
- WEINSTEIN, Stephen, The multimedia Internet, New York : Springer Science + Business Media, 2005, ISBN 0-387-23681-3, 978-0-387-23681-0.
- STAMOU, Giorgos, KOLLIAS, Stefanos, Chichester, England: John Wiley and Sons, 2005, ISBN 0-470-85753-6, 978-0-470-85753-3.
- HALSALL, Fred, Multimedia communications: applications, networks, protocols and standards, New YorkLondonHarlow, England : Pearson Education : Pearson Education : Pearson Education, 2001, ISBN 0-201-39818-4.
- GONZALEZ, Rafael C., WOODS, Richard E., EDDINS, Steven L., Digital image processing using MATLAB, Upper Saddle River, New Jersey : Pearson Education, 2004, ISBN 0-13-008519-7, 978-0-13-008519-1.
- GOPI, E.S., Algorithm collections for digital signal processing applications using Matlab, Dordrecht : Springer, 2007, ISBN 978-1-4020-6409-8.
- MILIC, Ljiljana, Multirate filtering for digital signal processing: MATLAB applications, Hershey, PA : Information Science Reference, 2009, ISBN 978-1-60566-178-0.

MATHEMATICAL MODELING AND SIMULATION
Course Code: INFO 308
Type of course: compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Prof. Nicoleta Breaz, PhD
Seminar tutor: Prof. Nicoleta Breaz, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	48	4	Summer	Grade	6

COURSE AIMS:
The general aim related to this course consists in getting knowledge which helps the students to use the mathematical concepts together with a specific software to model phenomenon from various fields as medicine, physics, chemistry, economy, sociology, etc.. Thus, through this course, the students acquire not only the knowledge of basics mathematical modeling aided by software products but also, they become open minded regarding the interdisciplinary matter and hence they get competencies in the use of the theoretical basis of computer science and of formal models in solving specific problems from various fields.

ENTRY REQUIREMENTS:
There are no compulsory prerequisites but the following courses are useful:
1. Probability and mathematical statistics
2. Mathematical software
3. Numerical calculus
4. Differential and partial derivatives equations

COURSE CONTENTS:

I. Elements of mathematical modeling and simulation
1. Introduction
2. Process of mathematical modeling
3. Models obtained through the translation of the problem in mathematical language
4. Simulation techniques and random numbers

II. Models based on statistical techniques
1. Simple linear regression model
2. Polynomial regression model
3. Other simple regression models
4. Multiple linear regression models
5. Other multiple regression models
6. Dynamic models

III. Models based on optimization techniques
1. Elements of mathematical programming
2. Transportation problems
3. Problems related to production and stocking
4. Problems of mixtures (dietary optimization, alloy mixture optimization)
5. Problems of cutting-stock
6. Problems from games theory
7. Other optimization problems

IV. Deterministic models based on equations
1. Problems of populations' dynamic
2. Deterministic models in epidemiology
3. Deterministic models in physics

TEACHING METHODS:
Lecture, discussion, exemplification.

LEARNING OUTCOMES:
- Identifying the appropriate models and methods for solving real-life problems;
- Giving the interpretation of mathematical and computer science (formal) models;
- Using the simulation in the study of the behavior of developed models and evaluation of results;
- Embedding the formal models in specific applications in various domains.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Practical project – 50%; continuous assessment – 50%.

RECOMMENDED READING:

- E.A. Bender, An introduction to mathematical modeling techniques, Dover, New York, 2000
- N.Breaz, Mathematical modeling and simulation, theory and applications, Seria Didactică, Univ. "1 Decembrie 1918"Alba Iulia, (in printing)
- D. J. Higham, N. J. Higham, MATLAB Guide, 2nd edition, SIAM, 2005
- M. P. McLaughlin, A tutorial on Mathematical Modeling (www.causascientia.org/math_stat/Tutorial.pdf), 1999
- Cleve Moler – Numerical Computing in MATLAB, SIAM, 2005
- ***– Documentation for MathWorks Products, R2009a- http://www.mathworks.com/

PRACTICE FOR THE DEVELOPMENT OF THE BACHELOR'S THESIS
Course Code: INFO 309
Type of course: Compulsory
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Cucu Ciprian, PhD
Seminar tutor: Cucu Ciprian, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	48	4	Summer	Grade	6

COURSE AIMS:

Understanding main concepts regarding doing research in Computer Science, both in terms of designing and conducting research projects as well as in documenting them approapriately

ENTRY REQUIREMENTS:

-

COURSE CONTENTS:

A. Defining research for bachelor's degree
B. Methodologies, literature review and implementing applications
C. Writing the thesis

TEACHING METHODS:

Lecture, Power Point presentations, conversation, exemplification.

LEARNING OUTCOMES:

Students will be able to complete a quality bachelor's thesis, according to standards.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Continuous assessment.

RECOMMENDED READING:

- Eco, U. - "How to write a thesis", *MIT Press,* Cambridge, 2015

MACHINE LEARNING AND PATTERN RECOGNITION
Course Code: INFO 310
Type of course: Optional
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Adriana Birlutiu, PhD
Laboratory tutor: Adriana Birlutiu, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	48	4	Summer	Grade	6

COURSE AIMS:

- This course gives an overview of many concepts, techniques, and algorithms in machine learning, beginning with topics such as classification and linear regression and ending up with more recent topics such as ensemble methods, support vector machines, and Bayesian networks.
- The course will give the student the basic ideas and intuition behind modern machine learning methods as well as a bit more formal understanding of how, why, and when they work.

ENTRY REQUIREMENTS:

Artificial intelligence.

COURSE CONTENTS:

- Supervised learning. Unsupervised learning
- Linear regression
- Classification
- Decision trees
- Ensemble methods
- Artificial neural networks
- Bayesian learning
- Support Vector Machines
- Unsupervised learning
- Pattern recognition
- Feature selection

TEACHING METHODS:

Lecture, conversation, exemplification.

LEARNING OUTCOMES:

- identify the type of a learning problem;
- understand the internal structure of a learning algorithm;
- apply a learning algorithm;

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

Written exam – 50%; continuous assessment – 50%.

RECOMMENDED READING:

- Mitchell, T., *Machine Learning*, The McGraw-Hill Companies, Inc., 1997
- Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer-Verlag, 2013
- Christopher Bishop. *Pattern Recognition and Machine Learning*, Springer, 2006.
- David Mackay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

INTELLIGENT COMPUTATION- BIO-INSPIRED TECHNIQUES

Course code: INFO 312
Type of course: optional
Language of instruction: English/Romanian
Name of lecturer: Corina Rotar, PhD
Full time studies

Form of instruction	Number of teaching hours per semester	Number of teaching hours per week	Semester	Form of receiving a credit for a course	Number of ECTS credits allocated
Class	42	3	Summer	Grade	6

COURSE AIMS:

Currently there is a strong interest towards the development of intelligent software applications in various fields such as mobile phones, gaming industry, etc. Intelligent Computation discipline supports training of specialists in this direction, forming strategies and skills to apply intelligent algorithms where traditional methods are not effective.
Objectives:

- Develop the students' ability to design software that is dedicated for solving the difficult problems by exploiting evolutionary algorithms.
- Study of the algorithms that is based on natural paradigms.
- Skills for approaching the complex problems in terms of evolutionary algorithms.
- Analytical study of the advantages and disadvantages of traditional algorithms versus stochastic algorithms for optimization problems.

ENTRY REQUIREMENTS:

- Imperative and Procedural Programming
- Artificial Intelligence

COURSE CONTENTS:

1. Fundamentals of Intelligence Computation
2. Paradigm of Genetic Algorithms
3. Paradigm of Evolutionary Strategies
4. Genetic Programming. Evolutionary programming
5. Artificial Immune Systems
6. Particle Swarm Optimization Technique
7. Ants Colonies. Other natural paradigm
8. Application of evolutionary algorithms in optimization
9. Introduction to fuzzy logic. Fuzzy systems.
10. Introduction in Neural networks
11. Bio-inspired Computing and applications I
12. Bio-inspired Computing and applications II

TEACHING METHODS:

- Lecture, Cooperative learning, Discussion and survey, Team-based learning.

LEARNING OUTCOMES:

- Implementation of an *evolutionary algorithm* to solve either an optimization or an NP-hard problem.

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

- Final project (oral presentation) 100%

RECOMMENDED READING:

- Goldberg D.E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Inc., 1989.
- Bäck T., Evolutionary Algorithms in Theory and Practice, Oxford University Press, 1996
- Dumitrescu D., Lazzerini B., Jain L.C., Dumitrescu A., Evolutionary Computation, CRC Press, Boca Raton London, New York, Washington D.C., 2000

ADVANCED NETWORKING TECHNIQUES
Course Code: INFO 314
Type of course: Optional
Language of instruction: English tutoring available for Erasmus students
Name of lecturer: Remus DOBRA, PhD
Seminar tutor: Remus DOBRA, PhD
Full time studies

Form of instruction	Number of teachinghours per semester	Number of teachinghours per week	Semester	Form of receiving a credit for a course	Number of ECTS creditsallocated
Class	56	4	Summer	Grade	6

COURSE AIMS:
After completing the course, students will be able to:

- Calculate the distortions caused by analog / digital signal in non-uniform quantization conditions and to calculate the delays in communications systems.
- To choose and design a digital communications network
- To program the VPN, SAN, Bluetooth, ZigBee, IEEE 1451, ESP - nodeMCU
- To work with communications protocols

ENTRY REQUIREMENTS:

-

COURSE CONTENTS:

1. Digital communication elements. VS analog digital communications. Long distance transmission. Types of modulations (amplitude, frequency, digital). Brief introduction of fiber-optic communications
2. The concept of network systems and services. Definitions. Topologies (bus, star, tree, Point-to-point, peer-to-peer, LAN, MAN, WAN
3. Service infrastructures: LAN Local Area Network, VLAN Value Added Network; Core network services: the WLAN Wireless Local Area Network, VLAN- Virtual Local Area Network
4. Radio communication. Performance radio frequency bands. Radio communications standards. 802.11. Security in radio transmissions
5. CAN- Controller Area Network, WAN- Wide Area Network, GSM- Global System for Mobile Communications
6. Services data communications: PAN- Personal Area Network, SAN- Storage Area Network, PAN, Zigbee, 3G, 4G, Bluetooth, I-Mode, WAP, Wireless and packet switching;
7. Applications of networks for sensor systems.

TEACHING METHODS:
Lecture, conversation, exemplification.

LEARNING OUTCOMES:

- Explaining and interpreting the hardware and software specific structures required in the fields of computer programming, high-level languages and specific CAD techniques for achieving electronic modules, microcontrollers, computer systems architecture, electronic programmable systems.
- programming VPN, SAN, Bluetooth, Zigbee, IEEE 1451
- programming WiFi networks based on ESP-nodeMCU
- using different types of communication protocols

LEARNING OUTCOMES VERIFICATION AND ASSESSMENT CRITERIA:

- Written report regarding practical work applications

RECOMMENDED READING:

- Jochen H. Schiller, *Mobile Communications*, Addison Wesley, 2003
- William Stallings, *Wireless Communications and Networks*, Prentince Hall, 2005