Transilvania University of Brasov, Romania
Study program: Applied Computer Science

Faculty: Mathematics and Computer Science
Study period: 3 years (bachelor)

1st Year – 1st Semester

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Mathematical analysis	5	3	2	0	0

Course description (Syllabus): Relations (properties of a relation, equivalence relations, order relations, function as an example of a relation); Using axiomatic properties of the set of real number; Sequences and series (sequences and series of real numbers, sequences and series of functions); Determining the radius of convergence and interval of convergence of power series; Using the Taylor series expansions of function; First order and higher order partial derivatives of real-valued functions of several variables, the Schwarz theorem, the first order differential and the partial derivatives of a composite real function; Study extremes of the real differentiable functions of several variable; Constrained extrema, the method of Lagrange multiplier; Establish the nature and calculation of improper integrals; Calculation of integrals depending on a parameter; Calculation of line integral; Calculation of multiple integrals.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Basic algebra for computer science	5	2	2	0	0

Course description (Syllabus): Functions and relations, Binary operations, Free semigroups, Groups, Permutations, Rings and fields, Tropical geometry, Matrices, Vector spaces, Linear codes.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Fundamental algorithms	6	2	1	2	0

Course description (Syllabus): This course introduces fundamental concepts in algorithms, covering the building blocks of algorithm design. Topics include pseudocode, variable and constant usage, instructions, and visibility of variables (global/local) within an algorithm. The curriculum explores subprograms, recursion, and the analysis of algorithm efficiency, including computing complexity across various scenarios. Specific algorithmic categories such as search and sort algorithms, divide and conquer algorithms, greedy algorithms, dynamic programming algorithms, and text processing algorithms are discussed. The course also involves a comparative analysis of implementations and programming methods. Additionally, it goes into intelligent algorithms, providing a foundational understanding of artificial intelligence concepts through research and case studies on selected issues. Practical components include problem-solving seminars on fundamental algorithm topics and hands-on laboratory sessions for the implementation of discussed algorithms.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Fundamentals of Programming	5	2	0	2	0

Course description (Syllabus): Getting familiar with the C language; Generic programming using macros; Understanding procedural programming; Pointers. Memory allocation and management; Developing function-oriented C application

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Mathematical and computational logic	4	2	1	0	0

Course description (Syllabus): Binary, decimal, octal, hexadecimal number systems; Conversions among bases; Addition, subtraction, multiplication and division of base b numbers, Representation of integer numbers (Sign and magnitude, One's complement, Two's complement); Addition, subtraction and multiplication of two's complement numbers; Fixed point numbers; Floating Point Numbers. The IEEE 754 Floating Point Standard (Formats and Rounding), Properties of Boolean Algebra; Truth Table; Boolean Functions. The disjunctive normal form and conjunctive normal form; Minimization of Boolean functions: Veitch-Karnaugh Maps, Quine McCluskey's method; Combinatorial circuits; The Algebra of GF (2). The operational and function domains; Reed-Müller expansions; Generalized Reed-Müller expansions.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Academic ethics and integrity I	2	1	0	2	0

Course description (Syllabus): LaTeX – Overview, Logical organization of a document in LaTeX, writing text and mathematical formulas, graphing and writing algorithms, LaTeX packages and their use in writing, using colors and graphics processing, beamer document class, writing a scientific paper in LaTeX.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
English/German/French/Spanish language 1	2	1	1	0	0

Course description (Syllabus): The English/German/French/Spanish language course attempts at revising, consolidating and improving grammar points at advanced level. It also provides opportunities for students to practice grammar structures and express themselves during the seminars. To this end, each lecture presents a theoretical issue which synthesizes essential information from outstanding books in English/German/French/Spanish morphology, followed by a wide range of exercises which will be approached during the following seminar. The exercises are designed in such a way so that students can solve them both individually and in pairs, during the seminars. The aim of this course is to provide students with comprehensive grammar structures linked to the following issues: problem verbs, modal verbs, passive voice, determiners and pronouns and relative clauses.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Physical education and sport 1	2	0	2	0	0

Course description (Syllabus): This course introduces fundamental principles of physical fitness, health, and various sports, promoting active lifestyles and teamwork.

1st Year – 2nd Semester

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Linear algebra, analytical and differential geometry	5	2	2	0	0

Course description (Syllabus): Basic notions of linear algebra (vectors, calculus with vectors, dependence and independence linearity, bases, dimensions, linear applications, bilinear forms, and quadratic forms, scalar products and other products); Matrix representation from linear algebra (of vectors, linear applications, bilinear forms and quadratic forms, of vectors products); Basic notions of analytical geometry (point, line, conic, plane, quadric and its equations, frame, relative positions, angles, distances); To apply techniques from linear algebra in analytical geometry, the knowledge of basic elements from differential geometry of curves and surfaces; Explain the necessity for use specific techniques of linear algebra and analytical geometry in mathematics, physics, technique and informatics; Explain the

use of specific techniques of linear algebra and analytical geometry (in particular and in details); Matrix interpreting the abstract definitions from linear algebra; Mathematics significations interpreting of some calculated elements; Use theoretical knowledge for problem solving; Interpreting algebraic and geometric some notions related to informatics.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Computer systems architecture	5	2	1	1	0

Course description (Syllabus): This course provides an in-depth understanding of the fundamental structure and internal operation of computer systems. You'll explore essential hardware components such as the Central Processing Unit (CPU), memory units, and Input/Output (I/O) systems, including peripheral devices. The curriculum delves into the digital-logic level representation of data, logic circuits, and registers, illustrating how information is internally processed. Key topics also include the detailed organization of the CPU, instruction sets, addressing modes, and the execution of programs. Additionally, you'll be introduced to assembly language programming and advanced architectural concepts like multiprocessing, multi-computer systems, and parallel, concurrent, and distributed computing.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Object oriented programming	6	2	0	2	0

Course description (Syllabus): This course offers an in-depth exploration of advanced C++ features and the fundamental principles of Object-Oriented Programming (OOP). You will master dynamic memory management, templates, exception handling, operator overloading, and lambda expressions. The core of the course focuses on OOP concepts such as class declaration, constructors, destructors, inheritance (including multiple and virtual inheritance), polymorphism through virtual functions, and the use of friend and static members. Additionally, you will learn about C++ streams for efficient input/output operations, file processing, and C++ string manipulation.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Data structures	6	2	0	2	0

Course description (Syllabus): Data structures used for efficiently storing and manipulating data in computer programs: linear structures like stacks, queues, linked lists; hash tables; binary search trees, balanced trees, heaps; advanced data structures: quadtrees, point-region trees, kd-trees, persistent trees. The data structures are described in the context of their applicability, together with the main operations and their complexity. Description of some data structures from the STL C++ library. In the laboratory: implementation of the data structures, using the data structures for solving problems.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Operating systems	5	2	0	2	0

Course description (Syllabus): This course provides an in-depth understanding of the fundamental principles and intricate workings of operating systems. Students will explore OS goals, architecture, and essential resource management across processes, memory, I/O, and storage, alongside concepts like security, protection, virtualization, and distributed systems. The curriculum covers OS structures, services, user interfaces, system calls, and the boot process, with a significant focus on file systems, including their concepts, access methods, directory structures, allocation strategies (contiguous, linked, indexed), and recovery mechanisms. Process management is examined in detail, encompassing processes, CPU scheduling algorithms (FCFS, SJF, Round Robin, Priority, Multilevel Queue), context switching, inter-process communication (IPC), threads, concurrency, multi-core programming, and remote procedure calls (RPC). The course thoroughly addresses process synchronization, covering race conditions, critical sections, mutexes, semaphores, monitors, classic synchronization problems (e.g., bounded buffer, readers-writers, dining philosophers), and robust strategies for deadlock prevention, detection, and recovery.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
English/German/French/Spanish language 2	2	1	1	0	0

Course description (Syllabus): The main objective of English/German/French/Spanish language seminars from the second semester is that of consolidating and improving the English/German/French/Spanish language knowledge acquired by students up to that point. This time the focus is no longer grammar but students' ability to express themselves as fluently and as accurately as possible by means of using a wide range of vocabulary items. To this end special material providing them with texts and exercises covering various artistic, cultural, political and social issues will be used. Apart from the emphasis placed on developing students speaking skills, the seminars will also focus on students' ability to communicate in writing. They will be taught to develop and organize their ideas logically and coherently, while at the same time using the language correctly.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Physical education and sport 2	2	0	2	0	0

Course description (Syllabus): This course introduces fundamental principles of physical fitness, health, and various sports, promoting active lifestyles and teamwork.

2nd Year - 1st Semester

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Graph algorithms	5	2	0	2	0

Course description (Syllabus): This course offers an in-depth exploration of graph theory and its essential algorithms, providing students with the tools to model and solve complex problems. The curriculum begins with introductory concepts in graph theory, laying the foundation for understanding various graph structures and properties. Students will then delve into graph traversal techniques, including partial and full traversals, as well as topological sorting. The course covers the identification of connected components and strongly connected components, and explores the fundamental concepts of cycles and trees. A significant portion is dedicated to algorithms for finding minimum spanning trees. Further advanced topics include arborescences, algorithms for computing distances and shortest paths, and a thorough study of network flows. By the end of this course, students will possess a strong theoretical and practical understanding of graph algorithms, enabling them to apply these powerful techniques across various computational domains.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Formal languages and automata theory	5	2	1	1	0

Course description (Syllabus): Automata theory stands at the core of theoretical computer science, encompassing important concepts essential for various applications. This course explores the theoretical formalization and automata, providing insights into their role in programming language compilation, text editor construction, and network modeling. Participants will develop skills in working with formal elements intrinsic to theoretical computer science. Key concepts covered include Markov systems, grammars, Chomsky classification, automata, push-down automata, regular expressions, and various types of analysis within the compilation flow.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Programming environments and tools	5	2	0	2	0

Course description (Syllabus): This course begins with introductory elements, exploring the fundamental question of why object-oriented programming is significant. It explores the key concepts and principles of object-oriented

programming, focusing on specifying and implementing a class with a Java perspective. The course covers inheritance and polymorphism in object-oriented programming, providing insights from the Java standpoint. Additionally, the module addresses structured exception handling in object-oriented programming, the intersection of object orientation and generic programming in a Java context, and explores object-oriented flows and object serialization, offering a comprehensive understanding of these essential aspects.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Databases	5	2	0	2	0

Course description (Syllabus): This course offers a comprehensive exploration of database systems, covering the fundamental distinction between logical and physical data organization. Students will master data modeling techniques, including logical and physical database design, and understand the ANSI/X3/SPARC model. A core focus is placed on the Entity-Relationship (ER) model and practical design rules, guiding through the main steps of logical database design. The curriculum also examines various operating modes and their representation. Practical application is emphasized through a case study on property association management using SQL, and the course concludes with insights into developing systems with databases. By the end, students will have a solid theoretical and practical foundation in database design, implementation, and management.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Modern C++ applied in artificial intelligence	5	2	0	2	0

Course description (Syllabus): This course is centered around tracing the evolution of the C++ language. By mastering the introduced concepts, participants will gain the ability to craft modern C++ code characterized by improved safety, speed, and simplicity. The C++ standard has undergone a continuous evolution with the introduction of C++11, C++14, C++17, and C++20. These updates facilitate writing more concise code, eliminating memory leaks, and enhancing overall runtime performance. Additionally, certain modules of the course focus on providing a C++ perspective for implementing practical, project-oriented artificial intelligence components in hands-on labs.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Cloud-based application development	5	2	0	2	0

Course description (Syllabus): This course dives into comprehensive cloud application development, covering DevOps principles, CI/CD, and robust security practices. You'll explore containerization, distributed systems, and Infrastructure as Code. The goal is to master building scalable, resilient, and secure cloud solutions.

2nd Year – 2nd Semester

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Artificial intelligence	5	2	0	2	0

Course description (Syllabus): This course provides a practical and theoretical introduction to Artificial Intelligence, with a strong emphasis on implementation using Python. Students will begin with an overview of the AI field, understanding its core concepts and applications. The curriculum then delves into fundamental machine learning algorithms, starting with Linear Regression and Logistic Regression, essential for predictive modeling. We will explore the building blocks of neural networks, covering the Linear Perceptron and advancing to Multilayer Perceptrons. Further topics include specialized network architectures such as Radial Basis Function Networks and Adaptive Resonance Models. The course concludes with an in-depth study of Convolutional Networks. By the end, students will understand key AI algorithms and know how to implement them using Python.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Visual programming	5	2	0	2	0

Course description (Syllabus): This course introduces visual programming with a focus on the .NET Framework and .NET Core. Students will cover fundamental concepts like data types, arrays, strings, instructions, classes, and namespaces. A core component is Object-Oriented Programming (OOP) in C#, alongside advanced features such as delegates, events, structs, collections, and generic classes. The curriculum also includes exception handling, attributes, ADO.NET, LINQ to objects, threads, and working with data streams. Students will be able to develop modern applications using the .NET ecosystem.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Computer networks	5	2	0	2	0

Course description (Syllabus): Assimilating the basic concepts on which the computer networks; Learning some basic notions on current network technologies; Acquisition of necessary knowledge for distributed programming; Formation of required skills for computers network administration; Using Windows commands to manage computer networks; IP and MAC addresses; Using and programming various algorithms for computer networks; To establish a local network and an inter-network.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Probability and mathematical statistics	5	2	1	1	0

Course description (Syllabus): Fundamental knowledge of probabilities and mathematical statistics and their use in problem solving. Among the important elements of the content: Field of events, operations with events. Probability field: classical/axiomatic definition, Conditional probability. Independent events. Total probability formula, Bayes' formula. Probabilistic schemes (two-state hypergeometric scheme, multi-state hypergeometric scheme, Poisson scheme, binomial scheme, multinomial scheme, Pascal's scheme) Random variables and discrete probability laws. The distribution function associated with a discrete random variable.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Statistical inference in machine learning	5	2	0	2	0

Course description (Syllabus): Introduction in probabilities. The notion of probability. Conditional probabilities. Statistical independence. Monty-Hall problem. Expected values. Mean. Variability. Standard deviation. Distribution of sample means. Important distributions. Asymptotes. Law of large numbers. Central limit theorem. Confidence intervals. Confidence intervals T. Hypothesis testing. P-value. Power. Variation in type I error rate, variability, sample size and effect size. Multiple testing. Resampling. Correlation and linear regression. Linear regression model. Residuals. Inference with regression. Properties of multivariate regression. Categorical variables. Adjustments. Choice of regression model. Generalized Linear Model (GLM).

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Web and Mobile Application Development with Angular, .Net, and Android	5	2	0	2	0

Course description (Syllabus): This course provides a practical approach to building modern, full-stack applications for both web and mobile platforms. It covers front-end development with Angular, the creation of robust back-end services using .Net, and native Android application development.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Physical Education and Sport	2	0	2	0	0

Course description (Syllabus): This course introduces fundamental principles of physical fitness, health, and various sports, promoting active lifestyles and teamwork.

3rd Year – 1st Semester

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Verification and validation of software systems	5	2	0	2	0

Course description (Syllabus): This course introduces the key principles and practices of Software Verification and Validation. We'll start with an overview of V&V and a clear understanding of the software quality assurance process. The curriculum then covers practical ways to ensure software reliability, including software verification and software validation, with a focus on functional testing. Students will also learn about non-functional testing and various advanced testing techniques. Beyond running tests, we'll look at the strategy behind them with test process management, teaching how to plan and manage testing. The course finishes by exploring test automation and current trends in the field. Students will understand software quality assurance well and be ready to help build high-quality software.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Web technologies	5	2	0	2	0

Course description (Syllabus): This course covers building modern web applications using Microservice-oriented Architectures. It provides hands-on experience with Docker and Docker Compose for container management, and includes the development of Web Architectures using Spring/Spring Boot. Also covered is Orchestrating Web Services using Kubernetes for managing deployed applications. The course is highly practical, featuring projects where solutions are built using microservices, Docker, and Kubernetes. A project demonstrating a complete web application flow from start to finish is also included. Students will possess the skills to design, develop, and deploy scalable web applications.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Mobile application development	5	2	0	2	0

Course description (Syllabus): Object-oriented programming in the context of the Android platform. Development of graphical user interfaces for mobile devices Development of advanced mobile applications. Presentation of the Android platform. Designing and implementing Android applications in Java Designing and implementing Android interfaces. Presentation of relevant advanced Java language concepts Design and implementation of databases for Android applications.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Professional practice	5	0	0	0	8

Course description (Syllabus): This course is designed to provide students with essential practical professional experience. It encompasses various forms of hands-on engagement, including internships within external companies or institutions, practical work conducted in university research laboratories, or active participation in faculty-led projects. The overall coordination and evaluation of these practical activities are managed by a designated practice coordinator for each respective study program.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Digital image processing	5	2	0	2	0

Course description (Syllabus): This course introduces the fundamentals of digital image processing, exploring essential stages from data representation and visual enhancement to advanced segmentation and filtering techniques. It covers point operations, geometric transformations, spatial and frequency domain filtering, morphological operations, and methods for edge and texture detection. The objective is to provide a solid understanding and practical skills for manipulating and analyzing digital images.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Neural networks and Deep learning	5	2	0	2	0

Course description (Syllabus): This course goes into cutting-edge deep learning techniques, exploring foundational neural networks alongside specialized architectures such as Convolutional Neural Networks (CNNs) for image processing, Recurrent Neural Networks (RNNs) for sequential data, and the transformative power of Transformers. It also covers modern learning paradigms including Unsupervised Learning, Generative Adversarial Networks (GANs), Self-supervised and Semi-supervised Learning, and an introduction to Reinforcement Learning. The aim is to equip participants with a comprehensive understanding of advanced AI models and their applications.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Computer graphics	5	2	0	2	0

Course description (Syllabus): This course offers a comprehensive introduction to computer graphics, covering 2D geometric transformations using homogeneous coordinates and various 2D curve representations, including parametric, interpolation, Coons, Bezier, and B-Spline curves. It delves into fundamental computational geometry algorithms for orientation testing, segment intersection, point-in-polygon tests, polygon triangulation, and area calculation. Additionally, the course introduces fractal geometry, exploring classic fractals such as the Cantor set, Koch curve, Lévy C curve, Dragon curve, plant fractals, Sierpinski shapes, and the Mandelbrot set, alongside elements of Turtle graphics. The objective is to equip students with the theoretical understanding and practical skills necessary for working with graphical representations and geometric algorithms.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Logical and functional programming	5	2	0	2	0

Course description (Syllabus): New programming paradigms are studied, and a paradigm-specific programming language is introduced: Prolog (Logic Programming) and LISP (Functional Programming). Use of recursive mathematical models. Use of basic notions specific to writing code in Prolog. Use of basic notions specific to writing code in LISP.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Practical coordination for bachelor thesis	5	0	0	0	6

Course description (Syllabus): This course focuses on the supervised process of developing a bachelor thesis. It involves a designated coordinator who oversees the student's entire thesis elaboration journey, providing guidance and support. The final grade for this course is assigned by the individual coordinator, based on the student's engagement, progress, and the quality of work demonstrated throughout the thesis development.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Database management systems	5	2	0	2	0

Course description (Syllabus): Using SQL dialects: theory and hands-on work with: Relational Algebra constructions (select, project, join, set operations); SQL dialects for MySQL and Oracle; Relational Design Theory (relational design, functional dependencies, BCNF, 4NF); Unified Modelling Language for database design (modelling for relations); Indexes; Transactions; Isolation levels; Constraints; Triggers; Referential integrity; Design of a DBMS application project written in Python, using MySQL connector, as practical work.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Parallel programming	5	2	0	2	0

Course description (Syllabus): This course provides a comprehensive exploration of NVIDIA's GPU architectures, detailing global, shared, and local memory structures, and the CUDA computational model. Participants will acquire fundamental skills in CUDA programming, developing GPU-accelerated algorithms for vectors, matrices, fractals, and reduction operations. The curriculum also delves into practical applications of GPU programming in critical fields such as Artificial Intelligence and Bioinformatics, offering a solid foundation in high-performance computing for complex problem-solving.

Course title	No. of credits	Number of hours per week			
		course	seminar	laboratory	project
Software testing techniques	5	2	0	2	0

Course description (Syllabus): This comprehensive course introduces the essential principles and methodologies of software testing, covering its critical role in the development lifecycle, fundamental objectives, and the seven ISTQB principles. Students will explore the testing lifecycle, differentiate between functional and non-functional testing types, and delve into various testing levels including unit, integration, system, and acceptance testing. The curriculum also includes practical test design techniques such as black-box (equivalence partitioning, boundary value analysis) and white-box (statement, branch coverage), defect management, and an introduction to test automation. Special attention is given to testing in Agile and DevOps environments, alongside an overview of quality standards and best practices.