RUSSIAN - ARMENIAN UNIVERSITY

Master Program

SYSTEM PROGRAMMING

MODULE DESCRIPTION

Professional Cycle (64 ECTS credit points)

Continuous Mathematical Models

Lecturer: Prof. Dr. Garnik Karapetyan
Labor intensity: 3 ECTS, 108 academic hours
Form of final control: exam (graded)
Standard Studies Period: 2 years

Course description: A course aimed at the construction, simplification, analysis, interpretation and evaluation of mathematical models that shed light on problems arising in the physical and social sciences. Derivation and methods for solution of model equations, heat conduction problems, simple random walk processes, simplification of model equations, dimensional analysis and scaling, perturbation theory, and a discussion of self-contained modular units that illustrate the principal modeling ideas. Students will normally be expected to develop a modeling project as part of the course requirements.

Recommended literature:

1. G.I. Marchuk. Mathematical modeling in the environmental problem .- M., Nauka, 1982.
2. G.I. Marchuk. Methods of computational mathematics .- M., Nauka, 1989.
3. V. Volterra. Mathematical theory of the struggle for existence .- M., Nauka, 1976.
4. Yu.M. Svirizhev, D.O. Logofet. Stability of biological communities .- M., Nauka, 1978.
5. L.A. Petrosyan, V.V. Zakharov. Introduction to mathematical ecology .- M., Nauka, 1986.
6. O.V. Besov, V.P. Ilyin, S.M. Nikolsky. Integral representations of functions and embedding theorems.-M., Nauka, 1975.
7. V.S. Of the world. Equations of mathematical physics.-M., Nauka, 1971.
8. L. HERMANDER. Linear differential operators with partial derivatives. - M., Mir, 1965.
9. L. HERMANDER. Analysis of linear partial differential operators, vol.1 (Theory of distributions and Fourier analysis) .- M., Mir, 1986.
10. V.S. Vladimirov. Generalized functions in mathematical physics.-M., Nauka, 1979.

History and Methodology of Applied Mathematics and Informatics

Lecturer: Prof. Dr. Parkev Avetisyan
Labor intensity: 2 ECTS, 72 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: The course is designed for the future specialists in applied mathematics. It aims to show students the historical roots of their specializations and to unfold some methodological problems that anyone working in the field most frequently encounters. This is the major rationale for the course.

Foreign language

Labor intensity: 5 ECTS, 180 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: This course will provide instruction in academic and professional language skills for non-native speakers of English. Emphasis is placed on development of integrated language skills for use in studying a particular content area. Upon completion, students should be able to demonstrate improved language skills for participation and success within the particular topic area.

Modern problems of applied mathematics and computer science

Lecturer: Prof. Dr. Parkev Avetisyan
Labor intensity: 2 ECTS, 72 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: The course is designed for the future specialists in applied mathematics. It aims to show students the modern problems of their specializations and to unfold some methodological problems that anyone working in the field most frequently encounters. This is the major rationale for the course.

Discrete stochastic models

Lecturer: Dr. L. Kachaturyan
Labor intensity: 3 ECTS, 108 academic hours
Form of final control: exam (graded)
Standard Studies Period: 2 years

Course description: The main purpose of this course for muster students is to demonstrate on various examples the power and generous possibilities of probabilistic methods applied to problems from different areas of mathematics.

The course consists of three main parts. The first part contains preliminary facts from probability theory including general discrete probability model, model of sequence of independent trials and Markov scheme (Markov chains). All these material is accompanied by practical examples.
The second part is dedicated to application of probabilistic methods in graph theory, combinatorics, Ramsey theory and extreme theory of sets. We demonstrate on various examples how to use probabilistic methods to obtain deterministic assertions. Generally such statements are existence theorems, but we also demonstrate how probabilistic approach gives an opportunity to create (deterministic) algorithms of finding objects which existence are proved.
In the third part of this course the mathematical theory of communication (Shannon's information theory) is expound. The notion of entropy and its properties are considered. The source coding theorems are discussed both for uniform and variable-length coding. The Shannon's theorems of optimal information transmission in discrete channels without memory are proved.

Recommendedliterature:

1. Parzen, E. (1962) Stochastic Processes, Holden-Day.
2. Dodge, Y. (2006) The Oxford Dictionary of Statistical Terms, OUP.

Numerical methods and optimization

Lecturer: Prof. Dr. Yuri Hakobyan
Labor intensity: 4 ECTS, 144 academic hours
Form of final control: exam (graded)
Standard Studies Period: 2 years

Course description: Numerical methods are an important branch in Applied Mathematics. It aims at numerically solving all kinds of mathematical problems which arise from practical applications and can be modelled by different mathematical equations or inequalities, for example, linear or nonlinear differential equations and integral equations.
Course prerequisite:
Most fundamental: advanced calculus and linear algebra.
The course is focused on both numerical methods and numerical analysis. And mathematicalanalysis is needed for nearly every numerical method to be introduced. So students should bevery solid in analysis, and have a very good feeling and understanding of numerical methodsand rigorous mathematical reasoning.

Recommended literature:

1. V.M. Verzhbitsky. Fundamentals of numerical methods.-M .: Higher School, 2002.
2. D. Watkins. Fundamentals of matrix calculations. - Moscow: BINOM. Laboratory of Knowledge, 2006.
3. D.V. Beklemishev. Additional chapters of linear algebra. - Moscow: Nauka, 1983.
4. D. Kincaid and W. Cheney. Numerical Analysis: Mathematics of Scientific Computing.- Brooks / Cole Publishing Company, 1991.

Algorithms and information security

Lecturer: Prof. Dr. R. Tonoyan
Labor intensity: 4 ECTS, 144 academic hours
Form of final control: exam (graded)
Standard Studies Period: 2 years

Course description: 1.Recursive algorithms. 2. Computable functions. Solvable and enumerable sets. 3. Algorithmically insoluble problems. 4. The 10th problem of Hilbert. 5. Probabilistic algorithms. 6. Hashing. 7. Introduction to the theory of algorithms and examples. 8. Complexity of algorithms. 9. LinearProgramming.

Recommendedliterature:

1. William Stallings. Cryptography and Network Security, 7th edition, Pearson, 2016.

Modern Operating Systems

Lecturer: Prof. Rouben Topchyan
Labor intensity: 4 ECTS, 144 academic hours
Form of final control: exam (graded)
Standard Studies Period: 2 years

Course description: The foundations of the organization of modern multiprocessor operating systems are studied in the course. The concepts of the organization of interactions between parallel processes and questions connected with deadlocks, the issues of planning of the computer system processors loading, as well as memory management, the organization of file systems, and information security issues are examined. The course contains both a thorough familiarization with system calls that control flows, processes, objects of inter process communication, and existing utilities that facilitate the process of developing software in the environment of operating systems of various types. The principles of the organization of operating systems for computer networks in multiprocessor systems are being studied. A comparative analysis of the performance of different operating systems is carried out. Methods and features of programming in Windows, Linux and Unix-like operating systems are being studied.

Topics cover the following themes:

- interfacingsynchronousandasynchronousprocesses
- input/outputsystem, buffering
- interrupts
- general parallel processes using fork and join
- criticalsections, mutualexclusion

- othercoordinationprimitives
- segmentationandpaging
- virtualmemory
- storageallocationandsharing

Recommendedliterature:

1. TanenbaumA., WoodhallA. .OperatingSystems: Design and Development. (3rded.) 2005.
2. Stollins W. OperatingSystems: InternalsandDesignPrinciples. — Addison Wesley,2003.
3. TanenbaumA. ModernOperating Systems (6th ed.), Prentice-Hall, 2010.

Architecture of Modern Computer Systems

Lecturer: Prof. Gagik Sardaryan
Labor intensity: 4 ECTS, 144 academic hours
Form of final control: exam (graded)
Standard Studies Period: 2 years

Course description: In the course the various fundamental components of computer systems are explored. The operation of the computer CPU in detail is explained and some of the many variations on the basic CPU design in different systems are introduced. The fundamental group of instructions that make up the repertoire of the computer and a fetch-execute cycle are learned. The variations in instruction sets and memory addressing techniques that differentiate computers from one another and extend the flexibility of the basic architecture are considered. In the course various CPU architectures, memory enhancements, and CPU organizations that expand the processing power of the CPU are explored. Various techniques used to perform I/O operations are considered as well. The advantages of adding additional CPUs in the form of multiprocessing and of off-loading operations into additional processors built into I/O modules are explored. The workings of various peripheral devices and some of the interactions between the various components in the computer system are presented.

Topics cover the following themes:

- PerformanceMeasurement
- InstructionSetArchitecture
- Arithmeticand ALU Design
- CPU DesignandExecution
- PipeliningforIncreasedPerformance
- Memory: Cache, Main, Virtual
- I/O DevicesandProtocols

Recommendedliterature:

1. Patterson, D.A. and J.L. Hennesey. Computer Organization and Design: The Hardware/Software Interface, Second Edition, San Francisco, CA: Morgan Kaufman, 1998.

2. Tanenbaum, A. S. Structured Computer Organization. EnglewoodCliffs, NewJersey: Prentice-Hall,1979.
3. John L. Hennessy and David A. Patterson. Computer Architecture: A Quantitative Approach (3rd ed.). MorganKaufmann Publishers.2001.
4. Joseph D. Dumas II. Computer Architecture: Fundamentals and Principles of Computer Design. CRC Press, 2005.
5. Tanenbaum A., Austin T. Computer Architecture. (6th ed.), Prentice-Hall, 2014.

Principles of Database Systems

Lecturer: Prof. Manuk Manukyan
Laborintensity: 4 ECTS, 144 academichours
Form of final control: exam (graded)
Standard Studies Period: 2 years

Course description: The theoretical and practical aspects of the implementation of the concept of databases are being considered within the course. Entity-relationship and object definition models are used as formalism for database modeling. Problems of relational database scheme design are emphasized. Issues for object, object-relational, deductive and semistructured data models are considered as well. SQL and OQL (Object Query Language) are examined. The algebraic and logical approaches of the query language construction, active, semi-structured and deductive databases are examined in detail. The problems of database integrity constraints are considered in the context of the relational model. Theoretical considerations are accompanied by development of corresponding projects in well-known database systems which help to acquire the knowledge. A comparative analysis of different approaches in database systems is provided.

Topics cover the following themes:

- an introduction to the implementation of database systems;
- datastoringessentials;
- systemandindexstructures:
- querydevelopment;
- parallelmanagement;
- transactiondevelopment;
- securityandinformationintegration;
- semi-structured and deductive databases;
- distributeddatabases.

Prerequisites: Data Structures and Fundamental Algorithms, Discrete Mathematics, Programming languages, Basics of Database Systems.

Recommendedliterature:

1. Ricardo C.M. Database Systems: Principles, Design and Implementation. Prentice Hall PTR Upper Saddle River,1990
2. C. J. Date, An Introduction to Database Systems, Addison-Wesley, 2004.

3. H. Garcia-Molina, J. D. Ullman, J. Widom, Database System: The Complete Book, Printice Hall, 2002.
4. L. A. Kalinichenko, Data model transformation method based on axiomatic data model extension, 4th International Conference on VLDB, pp. 549-555, Germany, September, 1978.

Intelligent Data Analysis

Labor intensity: 3 ECTS, 108 academic hours
Form of final control: exam (pass/fail)
StandardStudiesPeriod: 2 years

Course description: Intelligent data analysis is the process of detecting usable information in large data sets. In the intellectual analysis of data the methods of mathematical analysis are used to identify the patterns and trends existing in the data. Typically, such patterns can not be detected with traditional data viewing, because the links are too complex, or because of the excessive amount of data.The course educates students to a foundation level onbig data and the state of the practice of analytics. The course provides also an introduction to bigdata and a Data Analytics Lifecycle to address business challenges that leverage big data. Itprovides grounding in basic and advanced analytic methods and an introduction to big dataanalytics technology and tools, including MapReduce and Hadoop. The course has extensivelabs throughout to provide practical opportunities to apply the methods and tools taught in the course in the context of the Data Analytics Lifecycle.
Upon completingthe course, students will have the knowledge and practical experience to immediatelyparticipate effectively in big data and other analytics projects.

Topics cover the following themes:

- Introduction to Big Data Analytics
- Overview of Data Analytics Lifecycle
- Introduction to Data Mining
- Using R for Initial Analysis of the Data
- Theory and Methodsfor Big DataAnalytics and Statistical Modeling
- Technology and Tools for Big Data Advanced Analytics

Recommended literature:

1. Tan, Pang-Ning; Steinbach, Michael; and Kumar, Vipin (2005); Introduction to Data Mining, ISBN 0-321-32136-7
2. Feldman, Ronen; Sanger, James (2007); The Text Mining Handbook, Cambridge University Press, ISBN 978-0-521-83657-9
3. Guo, Yike; and Grossman, Robert (editors) (1999); High Performance Data Mining: Scaling Algorithms, Applications and Systems, Kluwer Academic Publishers
4. Han, Jiawei, MichelineKamber, and Jian Pei. Data mining: concepts and techniques. Morgan aufmann, 2006.
5. Torgo L. (2011). Data Mining with R. Learning with Case Studies. Minnesota: CRC Press.

Semantics of Programming Languages

Lecturer: Prof. Dr. Semyon Nigiyan
Labor intensity: 3 ECTS, 108 academic hours
Form of final control: exam (graded)
Standard Studies Period: 2 years

Course description: The course introduces the concept of basic semantics for procedural, functional, and logical programming languages. The bases of the untyped lambda-calculus, the Church-Rosser theorem, and the fixed-point theorem are being studied. The definition of the basic semantics for untyped functional programs is given. The programming methods in Lisp language are outlined. The problems of encoding of S-expressions and basic functions of the Lisp language by lambda-terms, ways of recording Lisp programs in the form of systems of equations in the untyped lambda-calculus, and a method for representing the basic semantics of the functional programs of the lambda-term are considered.
In the course, different paradigms of programming are generalized: structural, functional, logical. The course creates prerequisites for a deeper study of functional and logical programming languages.

Topics cover the following themes:

- Untypedλ-terms, β-redex, β-normal form, β-reduction, β-equality.
- Cherch-Rosser theorem, Corollary.
- Theorem of fixed pointcombinator.
- Untyped functional programs, Basic semantics.
- Lisp programming language.
- Basic semantics of Lisp-programs.
- Programming in Lisp.

Recommendedliterature:

1. Barendregt H. The Lambda Calculus. Its Syntax and Semantics. // North-Holland Pub. Co., 1981.
2. Henderson P. Functional Programming. Application and Implemention. // Prentice-Hall Int., 1980.
3. Field A.J., Harrison P.G. Functional Programming. // Addison-Wesley Pub. Co., 1988.
4. Nigiyan S.A., Avetisyan S.A. Semantics of Untyped Functional Programs. // Programming and Computer Software, Vol. 28, N3, 2002, p. 119-126.

Logical Programming

Lecturer: Prof. Dr. Semyon Nigiyan

Labor intensity: 2 ECTS, 72 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: The course is devoted to the logical programming paradigm. The course discusses logical programs and queries to them. The rules of the SLD-resolution are studied, the concepts of logical and procedural semantics are introduced. Questions of completeness and consistency of procedural semantics are considered. The concept of semantics of a fixed point, its completeness and consistency is studied. The foundations and methods of programming in the PROLOG programming language are outlined.
The course is associated with courses on the theory of algorithms and mathematical logic.

Topics cover the following themes:

- Variables, terms, atoms, formulas. Herbrand interpretations.
- Unification algorithm. Unification theorem.
- Logical programs, Queries. Logical semantics.
- SLD-resolution. Procedural semantics.
- Fixed point semantics.
- Prolog programming language.
- Programming in Prolog.

Recommended literature:

1. Lloyd J.W. Foundations of Logic Programming. // Springer-Verlag, 1984.
2. Clocksin W.F., Mellish C.S. Programming in Prolog. // Springer-Verlag, 1984.
3. Bratko I. Prolog Programming for Artificial Intelligence. // Addison-Wesley Pub. Co. 1986.
4. Nigiyan S.A. Horn Programmingwith Built-In Predicates. // Programming and Computer Software, Vol. 22, N1, 1996, p.19-25.

XML and Database

Lecturer: Prof. Manuk Manukyan
Labor intensity: 3 ECTS, 108 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: Theoretical and practical issues of supporting XML databases are the main subjects of the study in this course. The XML data model cause some interest because, unlike semi-structured data models, it supports the concept of database schemas and, unlike the inflexible schemes used in conventional data models, allows for declarations of more flexible database schemas. The course discusses the creation of XML database schemas based on XML Schema and OpenMath formalisms. Different approaches for creation of XML applications (considering these as DDL for databases) and query languages for XML are considered in the course. A special place is occupied by questions of studying the XQuery language.

Topics cover the following themes:
- XML language and XML data model,
- XML Scheme - typed XML,
- OpenMath - a language for representation and exchange of mathematical objects on the Web,
- XPath and XQuery. XML query languages.
- Futures and open problems.

Prerequisites: Database Systems Basics and Database Systems Principles.

Recommended literature:

1. Ronald Bourret, XML and Databases, 2005
2. Nicola, Matthias (28 September 2010). "5 Reasons for Storing XML in a Database". Native XML Database. Retrieved 17 March 2015.
3. Feldman, Damon.. Moving from Relational Modeling to XML and MarkLogic Data Models. MarkLogicWorld. Retrieved , 2015.
4. Bhargava, P.; Rajamani, H.; Thaker, S.; Agarwal, A. XML Enabled Relational Databases, Texas, The University of Texas at Austin, 2005
5. Elliotte Rusty Harold. The State of Native XML Databases, 2007.

Fuzzy Sets and Fuzzy Logic

Lecturer: Prof. Vachagan Vahradyan
Labor intensity: 2 ECTS, 72 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: Fuzzy sets are a generalization of ordinary sets. Mathematically this is expressed in the generalization of the characteristic function, which in the theory of fuzzy sets is called the membership function. It is defined on a universal set and takes values from the interval [0,1]. The theory of fuzzy sets and fuzzy logic allow us to formalize the individual, subjective attitude of a person to the studied subject and make decisions under conditions of uncertainty. In the course questions of representation of concepts of a subject domain in terms of linguistic variables and terms, construction of fuzzy rules of a conclusion in terms of fuzzy logic are considered. The basic algorithms of logical inference in fuzzy logic systems are being studied.

Topics cover the following themes:
- basic definitions and characteristics of fuzzy sets;
- the membership functions and the methods of their construction;
- operations on fuzzy sets;
- fuzzy relationship and operations on them;
- elements of fuzzy logic;
- fuzzy algorithms;
- elements of fuzzy geometry.

Recommended literature:

1. Zadeh L. A. Fuzzy sets. Information and Control, T. 8, № 3,1965.
2. D. Dubois and H. Prade .Fuzzy Sets and Systems. Academic Press, New York, 1988.
3. George J. Klir; Bo Yuan .Fuzzy sets and fuzzy logic: theory and applications. Prentice Hall, 1995.
4. Seising, Rudolf. The Fuzzification of Systems. The Genesis of Fuzzy Set Theory and Its Initial Applications—Developments up to the 1970s. Berlin, New York, Springer 2007.
5. Hans-Jürgen Zimmermann. Fuzzy set theory and its applications (4th ed.). Kluwer, 2001.

Neural networks

Lecturer: Prof. VachaganVahradyan
Labor intensity: 2 ECTS, 72 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: The theory of artificial neural networks (ANN) is one of the modern lines of computational mathematics and programming. Unlike the von Neumann programming ideology, which simulates conscious human computational actions, ANN simulates unconscious computational processes occurring in the brain of humans and animals. ANN is used to solve problems that are very difficult to algorithmize with the help of the von Neumann ideology. The course deals with the principles of modeling a biological neuron, the types of artificial neural networks and the methods of their training, the main classes of problems solved with the help of ANN.
The knowledge acquired through the course is used in the courses Big Data and DataMining and courses in machine learning.
Topics cover the following themes:

- architecture of high performance computers
- classification of multiprocessor computer systems
- principles for developing parallel algorithms and programs
- tools for developing parallel programs
- the parallel programming language OKKAM
- OpenMP technology for developing parallel programs for shared memory systems
- development of parallel programs for distributed memory systems using the MPI technology

Recommended literature:

1. Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J. Parallel Programming in OpenMP. - Morgan Kaufinann Publishers, 2000
2. Geist G.A., Beguelin A., Dongarra J., Jiang W., ManchekB., Sunderam V. PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Network Parallel Computing. MIT Press, 1994.
3. Group W, Lusk E, Skjellum A. Using MPI. Portable Parallel Programming with the Message-Passing Interface. - MIT Press, 1994
4. Kumar V., Grama A., Gupta A., Karypis G. Introduction to Parallel Computing.The Benjamin/Cummings Publishing Company, Inc., 1994

5. Pacheco, S. P. Parallel programming with MPI. Morgan Kaufmann Publishers, San Francisco. 1997.

Software Testing and Quality Assurance

Lecturer: Prof. Anna Hovakimyan
Labor intensity: 2 ECTS, 72 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: The course discusses the issues of software quality standardization. An overview of approaches and procedures aimed at ensuring the quality featuresof software is given. The process of softwaretesting as an integral and important part of the software control and qualityassuranceis presented. The course outlines methods for testing software throughout the software life cycle and covers software testing at the unit, module, subsystem, and system levels.The techniques for functional and structural testing and formal methods for programsverifyingare described. The tools for software testing are considered.
Topics cover the following themes:

- standards of software quality;
- automatic and manual techniques;
- generating and validating test data;
- static vs. dynamic analysis;
- functional testing;
- structurall testing;
- reliability assessment;
- programs verification.

Recommendedliterature:

1. W.Lewis. Software Testing and Continuous Quality Improvement. «Auerbach», 2000
2. J.D.McGregor, D.A.Sykes. A Practical Guide to Testing Object-Oriented Addison-Wesley, 2002г.
3. CemKaner, Jack Falk, Hung Q. Nguyen. Testing Computer Software, 2nd Edition , 2000.
4. Glenford J. Myers. The Art of Software Testing. (2nd ed.)1979.

Cloud Computing

Labor intensity: 3 ECTS, 108 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: The course covers the issues concerning cloud deployment and service models, cloud infrastructure, and the key considerationsin migrating to cloud computing. The course provides the required technology essentialsacross all domains—including server, storage, networking, applications, and databasesto helpdevelop a strong understanding of virtualization and cloud computing technologies.

Upon completion of this course, the student should be able to understand the transition from a Classic Data Center environment to Virtual Data Cente, virtualization at each layer – compute, storage, network, desktop, and application.
The course provides basic information about the emergence, development and use of cloud computing technologies.The models of cloud deployment and the cloud service provisioning model (SaaS, PaaS, IaaS, XaaS) are considered. The main advantages and disadvantages of cloud computing models and the solutions proposed on their basis are considered. The review of decisions of leading vendors - Microsoft, Amazon, Google is made. Methods for developing Web applications for deployment in the cloud, moving existing applications to itare being studied. Issues of security, scalability, deployment, backup and recovery in the context of the cloud infrastructure are being considered.

Topics cover the following themes:

- Classic Data Center
- Cloud computing basics
- Virtualization principles(compute, storage, networking, desktop, application)
- Business Continuity
- Cloud Computing Primer
- Cloud Infrastructure and Management
- Cloud Security
- Cloud Migration

Recommendedliterature:

1. Chou, Timothy. Introduction to Cloud Computing: Business & Technology. 2013
2. Ray J Rafaels *Cloud Computing: From Beginning to End.2014*
3. Hu, Tung-Hui A Prehistory of the Cloud. MIT Press, 2015.

Parallel Architectures and Parallel Programming Languages

Labor intensity: 3 ECTS, 108 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: The current state and development of parallel computing architectures, parallel programming tools in such areas as application of mathematical modeling methods in scientific and applied research, signal processing problems, in large information retrieval systems and database systems makes it urgent to familiarize and educate students about the architecture of parallel computing devices as well as the problems of parallel computing, the tools and methods of parallel programming.
Topics cover the following themes:

- architecture of high performance computers
- classification of multiprocessor computer systems

- principles for developing parallel algorithms and programs
- tools for developing parallel programs
- the parallel programming language OKKAM
- OpenMP technology for developing parallel programs for shared memory systems
- development of parallel programs for distributed memory systems using the MPI technology

Recommendedliterature:

1. Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J. Parallel Programming in OpenMP. - Morgan Kaufinann Publishers, 2000
2. Geist G.A., Beguelin A., Dongarra J., Jiang W., ManchekB., Sunderam V. PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Network Parallel Computing. MIT Press, 1994.
3. Group W, Lusk E, Skjellum A. Using MPI. Portable Parallel Programming with the Message-Passing Interface. - MIT Press, 1994
4. Kumar V., Grama A., Gupta A., Karypis G. Introduction to Parallel Computing.The Benjamin/Cummings Publishing Company, Inc., 1994
5. Pacheco, S. P. Parallel programming with MPI. Morgan Kaufmann Publishers, San Francisco. 1997.

Distributed Programming

Labor intensity: 2 ECTS, 72 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: The course examines the basics of creating distributed applications using C/C++/C# programming languages. The software components designed to work together in a networked environment are being described. Methods for implementing multi-agent systems, ways of using RPC protocols for calling remote procedures, and the cluster computation library MPI and PVM are being studied. The program components of the .NET platform are being presented to create multitasking and multithreaded applications.The course uses concepts and approaches studied in courses on parallel architectures and parallel programming languages and courses on operating systems.It is necessary to know the basics of creating sequential and parallel programs, to be able to create programs in the languages of structural and object-oriented programming.

Topics cover the following themes:

- parallel and distributed computing
- properties of distributed systems
- Distributed algorithmic mechanism design
- Parallel algorithms in shared-memory model
- Parallel algorithms in message-passing model
- Distributed algorithms in message-passing model

Recommendedliterature:

1. Coulouris, G.; Dollimore J; Kindberg T.; Gordon B., Distributed Systems: Concepts and Design (5th Ed.). Boston: Addison-Wesley,2011.
2. Andrews, Gregory R. Foundations of Multithreaded, Parallel, and Distributed Programming, Addison–Wesley, 2000.
3. Ghosh, Sukumar,Distributed Systems – An Algorithmic Approach, Chapman & Hall/CRC, 2007.
4. Lynch, Nancy A. Distributed Algorithms, Morgan Kaufmann,1996.

System Programming in Linux

Labor intensity: 2 ECTS, 72 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: In the course the architecture of the operating system Linux and methods of their communication are being considered. The course will cover the basic tools for writing programs for Linux, as well as the main application interfaces related to process control, working with the file system, inter process communication and networking. Attention will also be paid to many aspects of program development, such as: the use of dynamic libraries, the organization of multi-threaded computing and shared memory, and obtaining extended information about the system.

Topics cover the following themes:

- the architecture of the operating system Linux
- components of Linux
- basic features of Linux
- hierarchical file system
- tools for developing parallel programs
- multi-threaded computing
- Compilation Process and Libraries
- GNU tools

Recommended literature:

1. Linux System Programming. Nutshell Handbook, O'Reilly, Group, 2013.
2. Robert Love. Linux System Programming.O'Reilly,2002.
3. Lusk E, Skjellum A. Using MPI. Portable Parallel Programming with the Message-Passing Interface. - MIT Press, 1994.
4. Pacheco, S. P. Parallel programming with MPI. Morgan Kaufmann Publishers, San Francisco. 1997.
5. W. Richard Stevens. Unix Network Programming, Volume 1: The Sockets Networking API. et al. Addison-Wesley, 2003.
6. Richard Stevens UNIX Network Programming, Volume 2: Interprocess Communications. W. Prentice Hall, 1998.

Standard OpenGL

Labor intensity: 1 ECTS, 36 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: The course introduces students to modern methods and approaches to computer graphics. To create an application using computer graphics, the OpenGL library is being studied. The basic graphic primitives for drawing arbitrary curves and figures are described, methods of creation of three-dimensional graphic objects and the graphic user interface are being studied. The ways of working with the keyboard and the mouse are being considered. The course is accompanied by practical exercises. The course covers additional chapters for the course "Computer Graphics".

Topics cover the following themes:

- 2D and 3D computer graphics
- sprite graphics
- vector graphics
- computer animation
- rendering
- architecture of OpenGL library

Recommendedliterature:

1. David Rogers. Procedural Elements for Computer Graphics. McGraw-Hill, 1998.
2. John Lewell. Computer Graphics: A Survey of Current Techniques and Applications. Van Nostrand Reinhold,1985.
3. Jeffrey J. McConnell Computer Graphics: Theory Into Practice. Jones & Bartlett Publishers, 2006.
4. R. D. Parslow, R. W. Prowse, Richard Elliot Green Computer Graphics: Techniques and Applications. 1969.
5. Ron Fosner. OpenGL Programming for Windows 95 and Windows NT. 2000.
6. Richard S. Wright, Benjamin Lipchak. OpenGL SuperBible. (3rd ed.) SAMS Publishing, 2000.

Economy and policy of transition

Lecturer: Prof. Dr. Armen Darbinyan
Labor intensity: 1ECTS, 36 academic hours
Form of final control: exam (pass/fail)
Standard Studies Period: 2 years

Course description: This course studies the economics of public policy towards the environment. We begin by examining the problem of market failure in the presence of externalities and public goods. Then, we consider the public policy responses to these market failures, including command-and-control regulations, tax and subsidy incentives, marketable pollution permits, voluntary programs, and information as regulation. We consider these policies in contexts such as local pollution, climate change, threats to biodiversity, environmental justice, international trade, and development. In addition, we learn how to measure the costs and benefits of pollution control. By the end of the semester, you will learn how economists think about environmental problems, understand the advantages and disadvantages of a range of environmental policies, be able to conduct a cost-benefit analysis, and have a complete economic analysis of an environmental problem.