QUALIFICATION CHARACTERIZATION OF MAJOR FIELD OF STUDY "INFORMATICS"

FOR "MASTER OF SCIENCE" DEGREE WITH
PROFESSIONAL QUALIFICATION
"MASTER OF SCIENCE IN INFORMATICS"
1 YEAR (2 SEMESTERS)

I. Requirements to professional qualities and competences of enrolled students

Students enrolled in this major field of study have to submit diplomas for completed higher education BSc degree or MSc degree in the following programs: program "Informatics" - professional field 4.6. "Informatics and Computer Science"; program "Computer Systems and Technologies", program "Communication Equipment and Technology" - professional field 5.3 "Communication and Computer Techniques"; program "Electronics" - professional field 5.2 "Electrical Engineering, Electronics and Automation", program "Mathematics" - professional field 4.5 "Mathematics" and program "Mathematics and Informatics" - professional field 1.3 "Pedagogy of Teaching..."; program "Applied Optical Technologies"; program "Metrology" - professional field 4.1 "Physics"; program "Modelling, Technologies And Management In The Sewing Industry" - professional field 5.1 "Mechanical Engineering", as well as similar programs of the same professional fields.

Rules and regulations for submitting documents and enrollment are determined by the Faculty of Natural Sciences and Mathematics.

II. Requirements to professional qualities and competences of students, completed this major field of study

Neofit Rilski South-Western University prepares qualified experts in Informatics that can apply their knowledge and skills in the area of science, culture, education and economics in Bulgaria and abroad.

After completion of "Master in Informatics" degree they can successfully realize themselves as: programmers, system and network administrators and designers, graphic designers, scientists, experts in hardware and software technologies.

At completion of Master of Science in Informatics degree, students obtain:

- profound knowledge in the area of Informatics;
- good preparation in the area of Informatics and Mathematics as well as solid practical skills conforming to modern European standards and requirements;
- formation of affinity and ability for independent research and design;
- basis for continuing education at PhD degree;
- good opportunities for realizing as experts in Bulgaria or abroad;
- thinking style and affinity to the quickly changing requirements of the information society.

III. Requirements to preparation of students completing this major field of study

Students completed MSc degree in Informatics have to possess following knowledge, skills and competences:

- to conduct independent research, to model real processes and make computer automation systems for information maintenance;
- to use mathematical models and software packages for solving real economic, engineering and management problems in continuous and discrete macrosystems;
- to take part in development of program products and packages;
- to adapt and introduce program products and systems;
- to solve various optimization problems.

IV. Professional development

Masters of this program can be successfully implemented as: programmers, system and network administrators and designers, graphic designers, researchers, experts in hardware and software technologies.

Qualification characterization of Major Field of study "Informatics" for MSc degree is a basic document that determines rules for developing the curriculum. This qualification characterization is conformed to legislation in the area of higher education in Republic of Bulgaria.

STRUCTURE OF THE CURRICULUM
Field of Study: Computer Science
Degree: Master of Science, Period of Study: 1 year (2 semesters)

First Year			
First Semester	ECTS credits	Second Semester	ECTS credits
Compulsory Courses		**Compulsory Courses**	
Neural Networks	6.5	Coding Theory and Information Security	6.0
Component-Oriented Software Engineering	6.0	Theory of Algorithms	4.0
		XML Programming	3.0
Mathematical Modeling of Discrete Structures and Processes	6.5	Optional 3 (Group III)	2.0
Optional 1 (Group I)	5.0	Written State Exam or Graduate Thesis Defense	15.0
Optional 2 (Group II)	6.0		
Optional Courses (first group) (select one course)		**Optional Courses (third group)** (select one course)	
High Performance Parallel Computer Systems		Knowledge Databases	
Fault-Tolerance Computer Systems		Practical Course in Server Administration	
Principles of Grid-Networks		Practical Course in Aspect-Oriented Design and Programming	
Training at IT Company (Institution)		Design of Information Systems with Client-Server Architecture	
Optional Courses (second group) (select one course)			
Digital Communications			
Modern Modeling and Design Languages – UML			
Multilayer Database Applications			
Theory, Algorithms and Technologies for Speech Recognition			
	Total 30		Total 30

TOTAL FOR 1 ACADEMIC YEAR: 60 CREDITS

NEURAL NETWORKS

Semester: 1

Course Type: 6

Hours per week/FS/SS: 3 lecture; 1 exercise week/SS

ECTS credits: 6.5

Course Status: Obligatory course in the Computer Science - M.S. curriculum

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements.
Neural networks can be trained to solve problems that are difficult for conventional computers or human beings.

Course Aims:
Students should obtain knowledge and skills for designing of the neural network.

Teaching Methods: lectures, demonstrations and work on project

Requirements/Prerequisites: **Assessment:** course project

Registration for the Course: *by request at the end of the current semester*

Registration for the Exam: coordinated with the lecturer and Student Service Department

References:
Basic Titles:

1. Anthony L. Caterini, Dong Eui Chang. (2018) Deep Neural Networks in a Mathematical Framework, Springer, Switzerland.
2. V. Alluru B. Rao., C++ Neural Networks and Fuzzy Logic, London IDG Books Worldwide, Inc. 1998.
3. Patricia Melin, Modular Neural Networks and Type 2 Fuzzy Systems for Pattern Recognition, 2012, Springer.
4. R.B. Macy. Pattern recognition with Neural networks in C++, CRC Press, 1994.

Abbreviation: FS: Fall Semester
SS: Spring Semester

COMPONENT-ORIENTED SOFTWARE ENGINEERING

Semester: 1

Course Type: lectures, lab exercises

Hours per week/FS: 2 lecture hour per week and 2 labs hours per week/FS

ECTS credits: 6.0 credits

Course Status: Compulsory Course in Master of Science Curriculum of Informatics

Course Description:
The basic principles for creating and using components in the development of software solutions are presented in the course. The topics to be discussed are as follows: programming fundamentals. Understanding the component library; introduction to component creation2. Introduction to component creation; object-oriented programming for component writers; creating properties; creating events; creating methods; using graphics in components; handling messages; making components available at design time; modifying an existing component; creating a graphic component; customizing a grid; making a control data aware; making a dialog box a component; extending the IDE;

Course Objectives:
The aim of the course is to teach students some of the basics in creating component-oriented software solutions, using visual design environments and event-oriented programming.

After completion of the course students should be able to:

- create and use different types of components in the development of software products

Teaching Methods: Lectures, demonstrations, work on project.

Requirements/Prerequisites: Needed basic knowledge of object-oriented programming. Desirable knowledge of visual design environments and event-oriented programming, such as RAD Studio or/and Visual Studio.

Assessment: Evaluating the student shall be carried out in the sixth grad scale. Current control is performed during the laboratory sessions during the semester through two courseworks, one control test and one course project (50% of final grade). Course ends with a written exam on the material according to the attached syllabus (50% of final grade). When shown a weak exam score, the student appears on the makeup exam and retain the information received from the course work assessment.

Registration for the Course: The course is compulsory and is not applied for its study.
Registration for the Exam: Coordinated with the lecturer and Student Service Department

References:

1. Embarcadero Technologies. (2017). Component Writer's Guide: Embarcadero Technologies. Retrieved from Embarcadero Technologies Web Site: docwiki.embarcadero .com/ RADStudio/Seattle/en/Component_Writers_Guide_Index.
2. John Barrow, Linda Miller, Katherine Malan, Helene Gelderblom. (2005). Introducing Delphi Programming: Theory through Practice 4th Edition. Publisher: Oxford University Press.
3. Danny Thorpe. (1996). Delphi Component Design Paperback. Publisher: Addison-Wesley.
4. Marco Cantu. (2003). Mastering Delphi 7. Publisher Sybex.
5. Marco Cantu. (2010). Delphi 2010 Handbook: A Guide to the New Features of Delphi.
6. Nick Hodges. (2015). More Coding in Delphi. Publisher: Nepeta Enterprises.

MATHEMATICAL MODELING OF DISCRETE STRUCTURES AND PROCESSES

Semester: 1
Cours Tipe: Lectures and tutorials
Hours per week/FS/SS: 3 lecture hours, 1 tutorial hours per week/FS
ECTS credits: 6.5 credits

Course Status: Obligatory course in the Informatics M.S. Curriculum period of study 2 years.

Short Description:
The 1970s ushered in an exciting era of research and applications of networks and graphs in operations research, industrial engineering, and related disciplines. Graphs are met with everywhere under different names: "structures", "road maps" in civil engineering; "networks" in electrical engineering; "sociograms", "communication structures" and "organizational structures" in sociology and economics; "molecular structure" in chemistry; gas or electricity "distribution networks" and so on.
Because of its wide applicability, the study of graph theory has been expanding at a very rapid rate during recent years; a major factor in this growth being the development of large and fast computing machines. The direct and detailed representation of practical systems, such as distribution or telecommunication networks, leads to graphs of large size whose successful analysis depends as much on the existence of "good" algorithms as on the availability of fast computers. In view of this, the present course concentrates on the development and exposition of algorithms for the analysis of graphs, although frequent mention of application areas is made in order to keep the text as closely related to practical problem-solving as possible.
Although, in general, algorithmic efficiency is considered of prime importance, the present course is not meant to be a course of efficient algorithms. Often a method is discussed because of its close relation to (or derivation from) previously introduced concepts. The overriding consideration is to leave the student with as coherent a body of knowledge with regard to graph analysis algorithms, as possible.
In this course are considered some elements of the following main topics;
Introduction in graph theory (essential concepts and definitions, modeling with graphs and networks, data structures for networks and graphs, computational complexity, heuristics).
Tree algorithms (spanning tree algorithms, variations of the minimum spanning tree problem, branchings and arborescences).
Shortest-path algorithms (types of shortest-path problems and algorithms, shortest-paths from a single source, all shortest-path algorithms, the k- shortest-path algorithm, other shortest-paths).
Maximum- flow algorithms (flow-augmenting paths, maximum-flow algorithm, extensions and modifications, minimum-cost flow algorithms, dynamic flow algorithms).
Matching and assignment algorithms (introduction and examples, maximum-cardinality matching in a bipartite graph, maximum-cardinality matching in a general graph, maximum-weight matching in a bipartite graph, the assignment problem).
The chinest postman and related arc routing problems (Euler tours and Hamiltonian tours, the postman problem for undirected graphs, the postman problem for directed graphs).
The traveling salesman and related vertex routing problems (Hamiltonian tours, basic properties of the traveling salesman problem, lower bounds, optimal solution techniques, heuristic algorithms for the TSP).
Location problems (classifying location problems, center problems, median problems).
Project networks (constructing project networks, critical path method, generalized project networks).

Course Aims:
Students should obtain basic knowledge in Mathematical modeling in discrete structures and skills for solving optimization problems for graphs and networks.

Teaching Methods: lectures, tutorials, individual student's work

Requirements/Prerequisites: Linear Algebra, Linear optimization

Assessment: 3 homework D1, D2, D3; 2 tests K1, K2 (project); written final exam

Rating: 0,2.(D1+D2+D3)/3 + 0,5.(K1+K2)/2 + 0,3(Exam)

Registration for the Course: by request at the end of the current semester (when is not obligatory course).

Registration for the Exam: coordinated with the lecturer and Students Service Department

References:

1. Mirchev,Iv., "Graphs". "Optimization algorithms for networks", Blagoevgrad,2001 (in Bulgarian).
2. Mirchev,Iv., "Mathematical programming", Blagoevgrad, 2000 (in Bulgarian).
3. Minieka, E., "Optimization Algorithms for Networks and Graphs, Marcel dekker, Inc., New York and basel, 1978 /Майника, Э.Алгоритмы оптимизации на сетях и графах, М., "Мир" p1981/.
4. Christofides, N., graph Theory. An Algorithmic approach, Academic Press lnc /London/ Ltd. 1975, 1997 /Кристофидес, Н. Теория графов.Алгоритмический подход, М., "Мир", 1978/.
5. Swami, M., Thulasirman, Graphs, Networks and Algorithms, John Wiley & Sons, 1981 /Сваами М., К. Тхуласирман. Графы, сети и алгоритмы, М., "Мир", 1984/.
6. Зайченко Ю. П., Исследование операций, Киев, "Выща школа", 1988.

Abbreviation: **FS**: Fall Semester
SS: Spring Semester

HIGH PERFORMANCE PARALLEL COMPUTER SYSTEMS

Semester: 1

Course Type: lectures, lab exercises

Hours per week/SS: 2 lecture hours per week and 1 labs hours per week/FS
ECTS credits: 5.0 credits

Course Status: Optional Course in Master of Science Curriculum of Informatics

The proposed curriculum is high performance parallel computer systems, their programming and functional model. Deals with the parallel information processing and computer systems work in real time.

Course Objectives:
This course aims to provide basic knowledge on modern computer architectures and systems. Study is the development of RISC architectures, type the last generations of processors Itanium, hyper wire technology, transport and parallel computer systems.

After completion of the course students should be able to:

- have knowledge about how to build high-performance computer systems, different architectures and mathematical apparatus used in their realization.

Teaching Methods: Lectures, demonstrations, work on project and teamwork.

Requirements/Prerequisites: Needed basic knowledge of computer architecture, discrete mathematics, operating systems, numerical methods and optimization, programming.

Assessment: Evaluating the student shall be carried out in the sixth grad scale. Current control is performed during the laboratory sessions during the semester through coursework (30% of final grade). Course ends with a written exam on the material according to the attached syllabus (70% of final grade). When shown a weak exam score, the student appears on the makeup exam and retain the information received from the course work assessment.

Registration for the Course: **Submitted an application to the academic department at the end of current semester.**

Registration for the Exam: Coordinated with the lecturer and Student Service Department.

References:
Basic Titles:

1. Rob Williams, Computer Systems Architecture: A networking approach, Addison Wesley, 2000
2. Stallings, Computer Organisation and Architecturer, Prentice Hall, 2002
3. Miles Murdocca, Principles of Computer Architecture, Addison Wesley, 1999
4. Mark Hill, Readings in Computer Architecture., Morgan Kaufman, 1999
5. Vincejnt Heuring, Computer Systems Design and Architecture, Benjamin Cummings, 1997
6. Gene Golub, James Ortega, Scientific Computing for Computer Scientists: An Introduction to Parallel Computing, Academic Press 1993
7. Dillep Bhandarkar, Alpha Architecture and Implementation, Butterworth-Heinemann, 1996
8. Michael Flynn, Computer Architecture: Pipeland and Parallel Processor Design, Jones and Barlett Publishers International 1995

FAULT-TOLERANCE COMPUTER SYSTEMS

Semester: 1
Course Type: lectures, lab exercises
Hours per week/SS: 2 lecture hours per week and 1 labs hours per week/FS
ECTS credits: 5.0 credits

Course Status: Optional Course in Master of Science Curriculum of Informatics

The proposed curriculum is considered the principles of fault-tolerant computer systems, computer networks and software. Deals with architecture, patterns of diagnosis, analysis of capacity and how to design and create Dependable systems. The course provides additional knowledge in applied activities of the modern master specialist in informatics.

Course Objectives:
This course aims to provide basic knowledge to build modern Fault-Tolerance Computer Systems.

After completion of the course students should be able to:

- Fault-tolerance computer systems design.
- Fault-tolerance computer systems analysis.

Teaching Methods: Lectures, demonstrations, work on project and teamwork.

Requirements/Prerequisites: Needed basic knowledge of computer architecture, discrete mathematics, operating systems, numerical methods and optimization, programming.

Assessment: Evaluating the student shall be carried out in the sixth grad scale. Current control is performed during the laboratory sessions during the semester through coursework (30% of final grade). Course ends with a written exam on the material according to the attached syllabus (70% of final grade). When shown a weak exam score, the student appears on the makeup exam and retain the information received from the course work assessment.

Registration for the Course: **Submitted an application to the academic department at the end of current semester.**

Registration for the Exam: Coordinated with the lecturer and Student Service Department.

References:
Basic Titles:

1. Коваленко, А.Е., Гула, В.В., Отказоустойчивые микропроцессорые системы.Техніка,1986, Украина
2. Авиженис А. Отказоустойчивость-свойство, обеспечивающее, постоянную работоспособность цифров ых систем. Тр. Ин-та инженеров по электротехнике и радио электронике, 1978, т.66 номер 10
3. Mine H. Hatayama K. Performance evaluation of faulttolerant computing system. Proceedings of the FTCS -9 1979
4. Sauer A.M. Schmitter E.J. The fault-tolerant microcomputer system BFS -Proceedings of the FTCS – 11, 1981

PRINCIPLES OF GRID-NETWORKS

Semester: 1
Course Type: lectures, lab exercises
Hours per week/SS: 2 lecture hours per week and 1 labs hours per week/FS
ECTS credits: 5.0 credits

Course Status: Optional Course in Master of Science Curriculum of Informatics

The proposed curriculum is considered the principles of the GRID network. Deals with the principles and essence of the GRID network, features a GRID architecture, applied technology and other tools of the GRID network.

Course Objectives:
This course aims to provide basic knowledge of infrastructure architecture and development of the GRID network.

After completion of the course students should be:

- have knowledge about how to build and use of GRID systems and their tools..

Teaching Methods: Lectures, demonstrations, work on project and teamwork.

Requirements/Prerequisites: Needed basic knowledge of computer architecture, discrete mathematics, operating systems, numerical methods and optimization, programming.

Assessment: Evaluating the student shall be carried out in the sixth grad scale. Current control is performed during the laboratory sessions during the semester through coursework (30% of final grade). Course ends with a written exam on the material according to the attached syllabus (70% of final grade). When shown a weak exam score, the student appears on the makeup exam and retain the information received from the course work assessment.

Registration for the Course: **Submitted an application to the academic department at the end of current semester.**

Registration for the Exam: Coordinated with the lecturer and Student Service Department.

References:
Basic Titles:

1. Grid Computing. Making the Global Infrastructure a Reality. Fran Berman, Geoffrey Fox, Antony Hey (ed.). Wiley, 2003.
2. The Grid: Blueprint for a New Computing Infrastructure, Ian Foster, Carl Kesselman (ed.). Morgan Kaufmann, 1999.
3. Condor ® Version 6.4.3 Manual (http://www.cs.wisc.edu/condor/manual/v6.4/)
4. The Globus Grid Project (http://www.globus.org/research/)
5. Legion – a Worldwide Virtual Computer (http://legion.virginia.edu/introduction.html)
6. Core Jini, W. Keith Edwards
7. Jini ™ Network technology (http://wwws.sun.com/software/jini/)
8. The Jxta solution to P2P (http://www.javaworld.com/javaworld/jw-10-2001/jw-1019-jxta.html)

TRAINING IN IT COMPANY (ORGANIZATION)

Semester: 1

Type of Course: Extracurricular occupation

ECTS Credits: 5.0 credits

Course Status: Elective course in Master of Science Curriculum of Informatics

Course description:

The course is designed for acquiring practical skills and habits and the acquisition of expertise through introduction and participation in the activities of companies and organizations who design, implement, deploy and use modern IT.

Objectives:

This course aims to bind the knowledge gained from university education with hands-on activities performed in different IT companies (organizations).

Methods of teaching: Work in a real work environment.

Pre-requirements: Basic knowledge of Informational Technologies, Operating Systems, Databases, Programming.

Assessment: report; journal of the conducted practical training;

Registration for the Course: by request at the end of the current semester

Registration for the Exam: coordinated with the lecturer and the Student Service Office

DIGITAL COMMUNICATIONS

Semester: 1
Course Type: lectures, lab exercises
Hours per week/FS: 2 lecture hour per week and 2 labs hours per week/FS
ECTS credits: 6.0 credits

Course Status: Optional Course in Master of Science Curriculum of Informatics

Course Description:
The course discuses the problems concerning design, building and application of Digital Communication networks. The lectures begin with introduction to Digital Communications, principles of building, historical development and their contemporary classification. Open system interconnection model of ISO is presented. Teaching course includes basic principles of building and functioning of PDH, SDH, ISDN, B-ISDN and ATM networks. The lectures on the most popular in the world computer network Internet present its basic characteristics, principles of functioning and application. The laboratory work helps to better rationalization of lecture material and contribute to formation of practical skills.

Course Objectives:
The aim of the course is to acquaint students with the basic principles, standards and tendencies of development in the field of Digital Communications. This will help them in future to professionally solve system tasks in the area of network communications.

Teaching Methods: Lectures, demonstrations, work on project.

Requirements/Prerequisites: Needed basic knowledge of communications technology, operating systems and computer architectures.

Assessment: Evaluating the student shall be carried out in the sixth grad scale. Current control is performed during the laboratory sessions during the semester through two courseworks, one control test and one course project (50% of final grade). Course ends with a written exam on the material according to the attached syllabus (50% of final grade). When shown a weak exam score, the student appears on the makeup exam and retain the information received from the course work assessment.

Registration for the Course: Submitted an application to the academic department at the end of current semester.

Registration for the Exam: Coordinated with the lecturer and Student Service Department

References:

1. Мерджанов П., Телекомуникационни мрежи, Нови знания, С., 2010 г.
2. Мирчев С., АТМ комуникации, Нови знания, С., 2001 г.
3. Христов В. Цифрови комуникации, ЮЗУ "Н.Рилски" –Благоевград, 2004 г.
4. Христов Х., Мирчев С., Основи на телекомуникациите, Нови знания, С., 2001 г.
5. Lee, R.; Chiu, M.; Lin, J. Communications Engineering:Essentials for Computer Scientists and Electrical Engineers, Wiley-IEEE Press, 2007 г.

Abbreviation:
FS: Fall Semester

MODERN MODELING AND DESIGN LANGUAGES - UML

Semester: 1
Type of Course: **lectures and tutorials in computer lab**
Hours per week – **2 hours lectures and 2 hours tutorials in computer lab** \FS
Credits Numbers: **6.0 credits**

Course Status: Elective course in curriculum of major Informatics. Master degree.

Course Description: The course is an introduction to object-oriented software engineering, especially an object-oriented analysis and design, using Unified Modeling Language (UML). It covers topics related to the modeling software processes and software development that reflect the different aspects of a software engineering. Topics related to the possibilities of reengineering already prepared solutions and the use of object-oriented approach in the design of databases are also discussed.

Course Objectives: This course aims to provide students with knowledge of the methods used in object-oriented analysis and design using UML.
The main tasks are related to obtaining good knowledge and technical skills about:

- Quality software design and development;
- Use of the state-of-art methods for software design;
- Use of object-oriented methods such as domain models, use case, activity diagrams, class diagrams, interaction diagrams (sequence and communication) etc.;

- Analyze and verify the software stability;
- Teamwork;
- The software projects development.

Teaching Methods: Lectures, demonstrations, work on project.

Requirements/Prerequisites: It is recommended to prepare all the courses "Programming and Data Structures", "Object-Oriented Programming", "Database" and “Software Engineering”.

Assessment: Evaluating the student shall be carried out in the sixth grad scale. Current control is performed during the laboratory sessions during the semester through one course project and several software tasks (50% of final grade). Course ends with a written exam on the material according to the attached syllabus (50% of final grade). When shown a weak exam score, the student appears on the makeup exam and retain the information received from the course work assessment.

Registration for the Course: Submitted an application to the academic department at the end of current semester.

Registration for the Exam: Coordinated with the lecturer and Student Service Department

References:

Basic titles:

1. J. Rumbaugh, I. Jacobson, G. Booch, UML – The Unified Modeling Language Reference Manual, 2nd Edition, Addison-Wesley, 2005.
2. D. Rosenberg, M. Stephens, Use Case Driven Object Modeling with UML - Theory and Practice, Apress, 2007
3. D. Mouheb, M. Debbabi, M. Pourzandi, L. Wang, M. Nouh, R. Ziarati, D. Alhadidi, C. Talhi, V. Lima, Aspect-Oriented Security Hardening of UML Design Models, Springer, 2015
4. M. Seidl, M. Scholz, C. Huemer, G. Kappel, UML @ Classroom - An Introduction to Object-Oriented Modeling, Springer, 2012
5. B. Dathan, S. Ramnath, Object-Oriented Analysis, Design and Implementation an Integrated Approach, 2nd Edition, Spring, 2015
6. C. Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process, 2nd Edition, Prentice Hall, 2008; https://www.utdallas.edu/~chung/SP/applying-uml-and-patterns.pdf
7. B. Meyer, Object-Oriented Software Construction, 2nd Edition, ISE Inc. Santa Barbara (California), 2016; https://sophia.javeriana.edu.co/~cbustaca/docencia/ POO-2016-01/documentos/Object%20Oriented%20Software%20Construction-Meyer.pdf
8. S. Chenoweth, C. Rupakheti, CSSE 374 – Software Design, Winter, 2013-14, Department of Computer Science and Software Engineering; http://www.rose-hulman.edu/class/csse/csse374/

Additional titles:

1. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, Object- Oriented Modeling and Design, Prentice Hall, 1991
2. D. Coleman, P. Arnold, S. Bodoff, C. Dollin, H. Gilchrist, F. Hayes, P. Jeremaes, Object-Oriented Development: The Fusion Method, Prentice Hall, 1994

3. J. Martin, J. Odell, Object-Oriented Methods: Pragmatic Consideration, Prentice Hall, 1996

Abbreviation:
SS: Spring Semester

MULTI-LAYER DATABASE APPLICATIONS

Semester: 1
Course Type: lectures, lab exercises
Hours per week/FS: 2 lecture hour per week and 2 labs hours per week/FS
ECTS credits: 6.0 credits

Course Status: Optional Course in Master of Science Curriculum of Informatics

Course Description:
The course teaches methods for developing client-server and multi-layer databases applications through object-oriented integrated development environments (IDEs) for visual design and event-oriented programming. Various aspects of design databases applications using various objects: a datasets, tfield objects and data bound controls. Developed different applications to access data depending on their architecture: client-server and multi-layer (client-application server-server). Students learn different technologies for data access by: ADO, ADO.NET, dbExpress, IBExpress, DataSnap, Cloud applications and others.

Course Objectives:
The course objective is to give students an idea of some of the main technologies used for developing client-server and multi-layer applications for databases and their methods of use.

After completion of the course students should be able to:

- use different technologies when developing client-server and multi-layer applications for databases with different architecture.

Teaching Methods: Lectures, demonstrations, work on project.

Requirements/Prerequisites: Needed basic knowledge of databases and object-oriented programming. Desirable knowledge of programming languages C + +, Object Pascal (Delphi) and C #.

Assessment: Evaluating the student shall be carried out in the sixth grad scale. Current control is performed during the laboratory sessions during the semester through two courseworks, one control test and one course project (50% of final grade). Course ends with a written exam on the material according to the attached syllabus (50% of final grade). When shown a weak exam score, the student appears on the makeup exam and retain the information received from the course work assessment.

Registration for the Course: Submitted an application to the academic department at the end of current semester.

Registration for the Exam: Coordinated with the lecturer and Student Service Department

References:

1. Embarcadero Technologies. Developing Database Applications: Embarcadero Technologies. Retrieved from Embarcadero Technologies Web Site. 2017.
2. Marco Cantu. Mastering Delphi 7. Publisher Sybex. 2003.
3. Marco Cantu. Delphi 2010 Handbook: A Guide to the New Features of Delphi 2010. 2010.
4. Mario Szpuszta, Ingo Rammer. Advanced .NET Remoting. Publisher: Apress; 2nd ed. 2005.
5. Bob Swart. Delphi XE DataSnap Development Essentials. Bob Swart Training & Consultancy. 2011.
6. Cary Jensen Ph,D. Delphi in Depth: ClientDataSets, Publisher CreateSpace Independent Publishing Platform. 2011.
7. Andrew Troelsen. Pro C# 5.0 and the .Net 4.5 Framework, Apress. 2012.
8. Tim Patrick. Microsoft ADO.NET 4 Step by Step. Publisher: Microsoft Press. 2010.
9. Xavier Pacheco. Delphi for .NET Developer's Guide. Publisher: Sams Publishing. 2004.

Abbreviation:
FS: Fall Semester

THEORY, ALGORITHMS AND TECHNOLOGIES FOR SPEECH RECOGNITION

Semester: 1

Course Type: lectures, lab exercises

Hours per week/SS: 2 lecture hour per week and 2 labs hours per week/SS

ECTS credits: 6.0 credits

Course Status: Optional Course in Master of Science Curriculum of Informatics

Course Description: In this course, the theoretical foundations of modern speech processing technologies will be discussed. Some speech recognition software and using them to Bulgarian speech recognition will be viewed.

Course Objectives: This course aims to provide the students with the knowledge and practical experiences for the modern technology of natural speech processing.
After the course completed, the students should know and understand:

- The methods of speech signal processing and retrieve their features.
- The methodology of the construction of a phonetic and language model in a given language.

Teaching Methods: Browsing the Web, work on coursework and essay.

Requirements/Prerequisites: The knowledge by the courses "Programming and Data Structures", "Object-Oriented Programming", "Database", "Discrete Mathematics", "Linguistics", "Pattern Recognition" and "Neural Networks", are necessary.

Assessment: Evaluating the student will be carried out by the six-point marking scale. The final assessment is in the form of a test that covers the whole teaching material including theoretical questions and practical cases. The final mark presents 50% of the final test and 50% of the mark of the course work.

Registration for the Course: Applied to the academic department at the end of current semester.

Registration for the Exam: Coordinated with the lecturer and Student Service Department

References:

1. Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, Spoken Language processing – A Guide to Theory, Algorithm, and System Development, Prentice Hall PTR, 2001
2. Xuedong Huang, Alex Acero, Hsiao-Wuen Hon (2001) Spoken Language processing – A Guide to Theory, Algorithm, and System Development, Prentice Hall PTR
3. Stephen E. Leinson (2005) Mathematical Models for Speech Technology, John Wiley & Sons
4. Wu Chou, Bing Hwang Juang (2003) Pattern Recognition in Speech and Language Processing, CRC Press
5. Joseph Keshet, Samy Bengio (2009) Automatic Speech and Speaker Recognition – Large Margin and Kernel Method, John Wiley & Sons
6. Lawrence Rabiner, Ronald Schafer (2010) Theory and Application of Digital Speech Processing, Prentice Hall
7. Daniel Jarefsky, James Martin (2008) Speech and Language Processing (2nd Edition), Prentice Hall
8. Dong Yu, Li Deng (2014) Automatic Speech Recognition: A Deep Learning Approach, Springer
9. James R. Lewis (2011) Practical Speech User Interface Design, CRC Press
10. Homayoon Beigi (2011) Fundamentals of Speaker Recognition, Springer
11. Willi-Hans Steeb (2005) Mathematical Tools in Signal Processing with C++ and Java Simulations, University of Johannesburg, South Africa
12. K. R. Rao, D. N. Kim, J. J. Hwang (2010) Fast Fourier Transform: Algorithms and Apllications, Springer
13. Р. Кралева (2019) Разпознаване на реч: Корпус от говорима детска реч на български език, ISBN: 978-954-00-0199-9, УП „Неофит Рилски“, Благоевград.
14. Data Exchange System, http://childes.psy.cmu.edu/
15. Praat: doing phonetics by computer, http://www.fon.hum.uva.nl/praat/
16. WaveSurfer, http://www.speech.kth.se/wavesurfer/
17. The International Phonetic Association, http://www.langsci.ucl.ac.uk/ipa/index.html

CODING THEORY AND INFORMATION SECURITY

Semester: 2

Form of the course: **Lectures/exercises**

Hours (per week): **3 hours lectures + 2 hours exercises per week, +1 summer semester**
Credits: 6.0 **credits**

Status of the course in the educational plan:
Obligatory course in the educational plan of the speciality Informatics, MS

Description of the course:
The course starts with introduction of the main notions and results from the algebra, combinatorics and probability theory. As addition to the regular course some good codes and constructions of codes are considered. The BCH codes are introduced and studied by the decoding algorithms of Piterson-Gorenshtein-Cirler, Forney, Berlekamp-Massey and Euclid, together with the MDS codes. The main notions of the convolutional codes are considered including decoding. Some schemes for electronic signature are considered as well as some methods for encryption by public and secret keys. The students are requested to work on a thesis on encryption by public key based on large primes.

Scope of the course:
Obtaining knowledge of the theoretical backgrounds and practical abilities for applications of the Coding theory, cryptography and data protection. Development of abilities for work with linear and nonlinear and convolutional codes over finite field with special emphasis of their algebraic and combinatorial properties. Studying the principles of the modern cryptography by public and secret keys and electronic signature.

Methods: lectures, seminars, discussions, practical exercises, work on a thesis, problems solving

Preliminary requirements: The students must have basic knowledge from the Number theory, algebra, combinatorics, probability theory.

Evaluation: permanent control during the semester (two written exams) and exam in the semester's end in two parts – defense of the thesis and answering theoretical questions.

Registration for the course: ***according to the educational plan***

Registration for exam: up to agreement with the teacher and the Educational Office

Literature:

1. Notices (www.moi.math.bas.bg/~peter)
2. Raymond Hill. A First Course in Coding Theory, Calderon press, Oxford, 1986.
3. Richard Blahut. Theory and Practice of Error Control Codes, Addison-Wesley Publishing Company Reading, Massachusetts, 1984.
4. Кларк Дж., Дж. Кейн, Кодирование с исправлением ошибок в системах цифровой связи, Пер. С англ., Москва, „Радио и связь", 1987.

THEORY OF ALGORITHMS

Semester: 2

Course Type: lectures and seminars

Hours per week/FS/SS: 3 lecture; 1 exercise week/FS
ECTS credits: 4**.0**

Course Status: Obligatory course in the Computer Science - M. S. curriculum

In this course
In this course will present some popular algorithms (sorting algorithms and so on) which have been precisely specified using an appropriate mathematical formalism--such as a programming language and we can analyze them:

- determine the running time of a program as a function of its inputs;
- determine the total or maximum memory space needed for program data;
- determine the total size of the program code;
- determine whether the program correctly computes the desired result;
- determine the complexity of the program--e.g., how easy is it to read, understand, and modify; and,
- determine the robustness of the program;

Course Aims:
Students should obtain knowledge and skills to write algorithms and compare the algorithms which solve the same problem;
.
Teaching Methods: lectures, demonstrations and work on project

Requirements/Prerequisites: Linear algebra, Computer languages.
Assessment: course project
Registration for the Course: *by request at the end of the current semester*
Registration for the Exam: coordinated with the lecturer and Student Service Department

References:
Basic Titles:

1. Mathworks. Programming in Matlab. New York. Pres Mathwork 2002
2. John Morris, *Data Structures and Algorithms, 1998* University of Western Australia;
3. Bruno R. Preiss, Data Structures and Algorithms with Object-Oriented Design Patterns in C++, University of Waterloo, Waterloo, Canada
4. Paul E. Black. Dictionary of Algorithms and Data Structures, http://www.itl.nist.gov/; Abbreviation:

FS: Fall Semester
SS: Spring Semester

XML PROGRAMMING

Semester: 2
Course Type: lectures and lab exercises
Hours per week/SS: 2 lectures and 1 lab hours per week / SS
ECTS credits: 3.0 credits

Course Status: Compulsory Course in Master of Science Curriculum of Informatics

Course Description:
The course introduces students to the fundamentals and development of the XML language and related syntactic features, such as a well-formed document, validation, hierarchical

structure, namespace, and others. Also detailed XML-related technologies (and XML-based languages) such as DTD, XML schematics, Relax NG, Schematron, DOM, xPath, XSLT, xQuery and others are also considered. Particular attention is paid to the relationship between XML and databases, and in particular the capabilities of the XML database management systems. The course also provides additional knowledge related to Event-Oriented Programming and XML, SAX usage, LINQ capabilities for XML, content distribution and external news, Web services and related technologies such as COM, DCOM, CORBA, XML-RPC, REST, and more

Course Objectives:
The aim of the course is to acquire in-depth knowledge of the basics of XML and related technologies and their application in the development of various business applications.

Teaching Methods: Lectures, demonstrations, work on project.

Requirements/Prerequisites: Studying the course requires students to have basic knowledge of programming and data structures, object-oriented programming, databases and database management systems. It is also desirable that students have also studied the disciplines related to visual design environments and event-oriented programming such as Delphi, C ++ Builder and Visual Studio.

Assessment: Evaluating the student shall be carried out in the sixth grad scale. Current control is performed during the laboratory sessions during the semester through two courseworks, one control test and one course project (50% of final grade). Course ends with a written exam on the material according to the attached syllabus (50% of final grade). When shown a weak exam score, the student appears on the makeup exam and retain the information received from the course work assessment.

Registration for the Course: The course is compulsory and is not applied for its study.

Registration for the Exam: Coordinated with the lecturer and Student Service Department

References:

1. Joe Fawcett, Liam R.E. Quin, Danny Ayers. Beginning XML, Fifth Edition. John Wiley & Sons, Inc. 2012.
2. Elliotte Rusty Harold. XML 1.1 Bible, 3rd Edition. Wiley Publishing, Inc. 2004.
3. Dorothy J. Hoskins. XML and InDesign, First Edition. O'Reilly Media, Inc. 2013.
4. Aaron Skonnard, Martin Gudgin. Essential XML Quick Reference: a Programmer's Reference to XML, XPath, XSLT, XML Schema, SOAP, and More. Pearson Education, Inc. 2002.
5. Doug Tidwell. XSLT, Second Edition. O'Reilly Media, Inc. 2008.
6. Priscilla Walmsley. XQuery, First Edition. Priscilla Walmsley. 2007.

KNOWLEDGE DATABASES

Semester: 2
Type of Course: **lectures**
Hours per week – **2 hours lectures /autumn semester**
Credits Numbers: **2**.0 **credits**

Course Status: Elective course in curriculum of major Informatics, Magister degree.

Course description:

The course is introduction in main aspects of knowledge bases and application.

Objectives:

The student should obtain knowledge of:

- Knowledge bases approach.
- Application of knowledge bases.

Methods of teaching: lectures, discussions, project based method.

Pre-requirements: Functional and Logical programming, Artificial Intelligence, and Mathematical Logics (core courses)

Assessment and Evaluation

Project- 50%
Final exam- 50%

Registration for the Course: by request at the end of the current semester

Registration for the Exam: coordinated with the lecturer and the Student Service Office

References:

1. Нишева, М., Д. Шишков, Изкуствен интелект, Изд. „Интеграл, Добрич, 1995
2. Ирена Атанасова, Създаване на експертни системи (Expert Systems Development), Издателство на ЮЗУ „Н. Рилски“, онлайн издание, 2018
3. Knowledge-Based Systems. Rajendra Akerkar , Priti Sajja, 2009 , ISBN10: 0763776475.
4. Engineering of Knowledge-Based Systems. Avelino J. Gonzalez, Douglas D. Dankel, Prentice Hall (2000), ISBN-10: 0130189731.
5. Expert Systems: Principles and Programming, Fourth Edition.Joseph C. Giarratano, Gary D. Riley, 2004, ISBN-10: 0534384471

PRACTICAL COURSE IN SERVER PROGRAMMING

Semester: 2
Course Type: lab exercises
Classes/FS/SS: 2 labs per week /SS
ECTS Credits: 2.0 credits
Course Status: Optional course in MSc Curriculum of Informatics
In this course are discussed the basic actions and problems related to network administration of Linux based systems. The course is aimed at providing the necessary skills needed to perform nearly all important administration activities required to manage a Linux/Windows network configuration, the basic setup and management of the most commonly used Internet services.
Course Objectives: The course is aimed at introducing to students the common concepts in network administration by discussing the basic activities regarding the administration of a Linux/Windows network configuration.
Teaching Methods: Labs, demonstrations, work on project and teamwork.
Requirements/Prerequisites: Needed basic knowledge of operating systems, programming, computer architectures, computer networks and communications.

Assessment: written final exam
Registration for the Course: a request is made by students at the end of the previous semester
Registration for the Exam: coordinated with lecturer and Student Service Department
References:

1. Олаф Кирх, Тери Доусън, Ръководство на мрежовия администратор.
2. Мат Уелш, Матиас Далхаймер, Ръководство за LINUX.
3. Алдениз Рашидов. Инсталиране и конфигуриране на Web сървъри под Linux и Windows (2012)
4. Microsoft SQL ServerNotes for Professionals book
5. Ronald Bardford. Effective MySQL Backup and Recovery (2012)
6. Shijimol Ambi Karthikeyan (2018) Practical Microsoft Azure IaaS: Migrating and Building Scalable and Secure Cloud Solutions Paperback
7. Gabriel N. Schenker (2018) Learn Docker - Fundamentals of Docker 18.x: Everything you need to know about containerizing your applications and running them in production, PACKT Publishing
8. Greg D. Moore (2016) IT Disaster Response: Lessons Learned in the Field, APress
9. Lawrence E. Hughes. The Second Internet: Reinventing Computer Networking with IPv6 (2010)
10. Raphaël Hertzog, Roland Mas. The Debian Administrator's Handbook (2012)
11. Ron Aitchison. Pro DNS and BIND 10 (2011)
12. Ronald Bardford, Chris Schneider. Effective MySQL Replication Techniques in Depth (2013)

PRACTICAL COURSE IN ASPECT-ORIENTED DESIGN AND PROGRAMMING

Semester: 2nd
Course Type: seminars and labs
Classes/WS/SS: 1 seminar hour and 1 hour labs per week/SS
ECTS Credits: 2.0 credits
Course Status: Optional course from the Computer Science Master Curriculum.
Short Description:
This course observes the advanced paradigm of programming – the Aspect oriented programming. In the course the basics of aspect oriented programming with AspectJ and Aspect C# is provided.
Course Aims:
The course aim is to give theoretical and practical background to students to use Aspect oriented languages and approaches in software development.
Teaching Methods: Labs.
Requirements/Prerequisites: Data Structures, Object Oriented Programming
Exam: final exam
Registration for the course: A request is made by students at the end of the current semester
Registration for the exam: Coordinated with lecturer and Students Service Department

References:

1. Kiczales, Gregor; John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingtier, and John Irwin (1997). "Aspect-Oriented Programming", Proceedings of the European Conference on Object-Oriented Programming, vol.1241, pp.220–242. The paper originating AOP.
2. Filman, Robert E.; Tzilla Elrad, Siobhàn Clarke, and Mehmet Aksit. Aspect-Oriented Software Development. ISBN 0-321-21976-7.
3. В. О. Сафонов, Аспектно-ориентированное программирование (http://www.vladimirsafonov.org/other/Safonov_AOP_2011_final.pdf)
4. Jacobson, Ivar; and Pan-Wei Ng. Aspect-Oriented Software Development with Use Cases. ISBN 0-321-26888-1.
5. Clarke, Siobhán; and Elisa Baniassad (2005). Aspect-Oriented Analysis and Design: The Theme Approach. ISBN 0-321-24674-8.
6. Matthew D. Groves (2013) AOP in .NET: Practical Aspect-oriented Programming, Manning Publications Company
7. Ramnivas Laddad (2009) AspectJ in Action: Enterprise AOP with Spring, Manning Publications Company
8. The AspectJTM 5 Development Kit Developer's Notebook, online: https://www.eclipse.org/aspectj/doc/released/adk15notebook/index.html
9. Pawlak, Renaud; Lionel Seinturier, and Jean-Philippe Retaillé. Foundations of AOP for J2EE Development. ISBN 1-59059-507-6.
10. Laddad, Ramnivas. AspectJ in Action: Practical Aspect-Oriented Programming. ISBN 1-930110-93-6.
11. Marijn Haverbeke (2018) Eloquent JavaScript Online: https://eloquentjavascript.net/Eloquent_JavaScript.pdf
12. Ivan Kiselev (2002) Aspect-Oriented Programming with AspectJ, Sams Publishing

DESIGN OF INFORMATION SYSTEMS WITH CLIENT-SERVER ARCHITECTURE

Semester: 2
Course Type: lectures, lab exercises
Hours per week/SS: 1 lecture hour per week and 1 lab hours per week/SS
ECTS credits: 2.0 credits

Course Status: Optional Course in Master of Science Curriculum of Informatics

Course Description:
The course includes basics of database management systems and related topics: introduction to the database management systems, requirements, architecture and basic principles of operation; comparison between the most widely used database management systems; basics of planning, installing, configuring and managing components of a DBMS and its instances; tools for working with database management systems, familiarization with the tools SQL Server Management Studio and IBConsole; design of relational databases and create a physical diagram of database scheme in the DBMS; create and modify tables in the DBMS, use types, expressions and functions; defining keys and restrictions when creating relationships between tables, creating and using indexes, working with diagrams in the DBMS; working with SQL statements INSERT, DELETE, and UPDATE with insert, delete and update data; working with the SQL statement SELECT retrieving data; working with

joins in extracting information from multiple tables, creating and using views; create and work with stored procedures in the DBMS, define custom functions; working with transactions and locks in the DBMS; create and use triggers in the DBMS; security system DBMS, working with logins, roles and users, authentication and authorization; exporting and importing data, DBMS capabilities for backup and restore databases;

Course Objectives:
Students should obtain basic knowledge and skills for database management systems.

Teaching Methods: Lectures, demonstrations, work on project.

Requirements/Prerequisites: Needed basic knowledge of databases.

Assessment: Evaluating the student shall be carried out in the sixth grad scale. Current control is performed during the laboratory sessions during the semester through two courseworks, one control test and one course project (50% of final grade). Course ends with a written exam on the material according to the attached syllabus (50% of final grade). When shown a weak exam score, the student appears on the makeup exam and retain the information received from the course work assessment.

Registration for the Course: Submitted an application to the academic department at the end of current semester.

Registration for the Exam: Coordinated with the lecturer and Student Service Department

References:

1. C. J. Date. An Introduction to Database Systems. Eighth Edition. Pearson. 2003.
2. Elmasri, R., Navathe, S. Fundamentals of Database Systems. Sixth Edition. Pearson. 2013.
3. C. J. Date. SQL and Relational Theory: How to Write Accurate SQL Code. Second Edition. O'Reilly Media. 2011.
4. A. Jorgensen, P. LeBlanc, J. Chinchilla, J. Segarra, A. Nelson. Microsoft SQL Server 2012 Bible. John Wiley & Sons, Inc. 2012.
5. O. Thomas, P. Ward, B. Taylor. Administering Microsoft SQL Server 2012 Databases. Microsoft Press. 2012.
6. P. Atkinson, R. Vieira. Beginning Microsoft® SQL Server® 2012 Programming. John Wiley & Sons, Inc. 2012
7. R. Dewson. Beginning SQL Server for Developers. Fourth Edition. Apress. 2015.

Abbreviation:
SS: Spring Semester